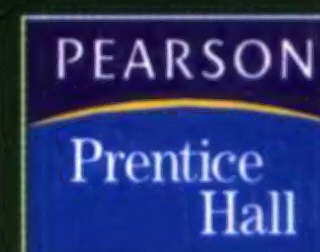
国外电子与通信教材系列



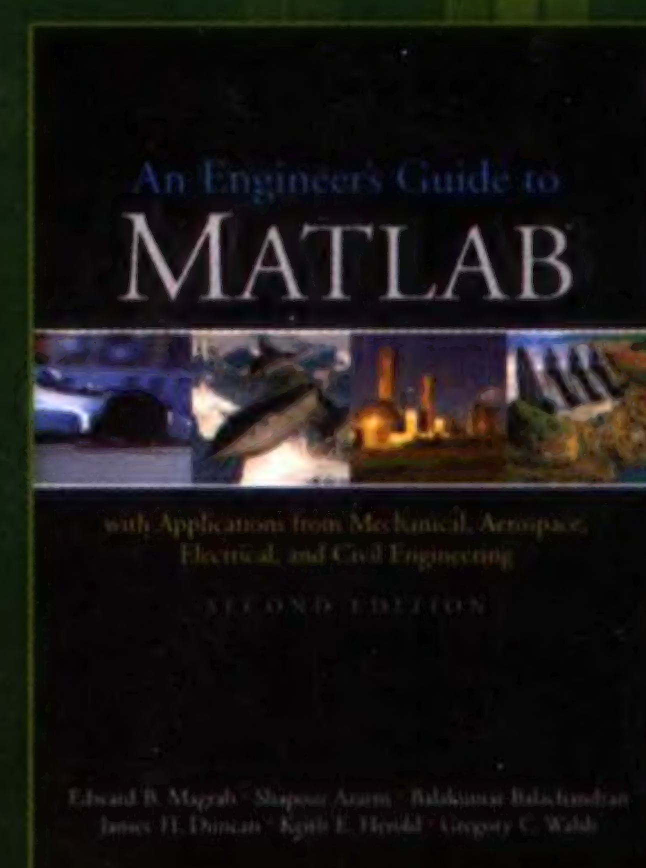
MATLAB
原理与工程应用
（第二版）

An Engineer's Guide to MATLAB
with Applications from Mechanical, Aerospace, Electrical, and Civil Engineering

Second Edition

[美] Edward B. Magrab Shapour Azarm Balakumar Balachandran James H. Duncan Keith E. Herold Gregory C. Walsh 著

高会生 李新叶 胡智奇 等译





電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

MATLAB原理与工程应用

（第二版）

这是一本优秀的MATLAB入门书籍。面向MATLAB的初学者，使他们通过阅读本书可以非常容易地掌握MATLAB的基本编程方法，并能继续提高MATLAB的运用能力。书中提供的大量典型实例简单易学，进一步充实了本书内容。作者在每章结尾列出的练习题非常好，不落俗套。

——Brittany Coats 和 Susan Margulies，宾夕法尼亚大学

书中的实例非常好，我对所提供的大量自定义函数尤其感到满意。书中给出的练习也是相当不错的。

——Bradley T. Burchett，Rose-Hulman 技术学院

本书的亮点是大量精彩实例和第8章到第14章的内容，这些内容与本书的前一部分相对独立。作为一本工程应用（尤其是机械工程应用）的书籍，本书具有很强的综合性，写得相当好。

——Vijay Sekhar Chellaboina,，Tennessee 大学

与我曾经读过的相关著作相比，本书有一定的深度，典型实例具有独到之处。本书的部分章节覆盖了机械零件设计、动力学与振动、控制系统、流体力学和热传导等多个学科领域，体现了这些领域的先进技术，提供了较好的工科教材。

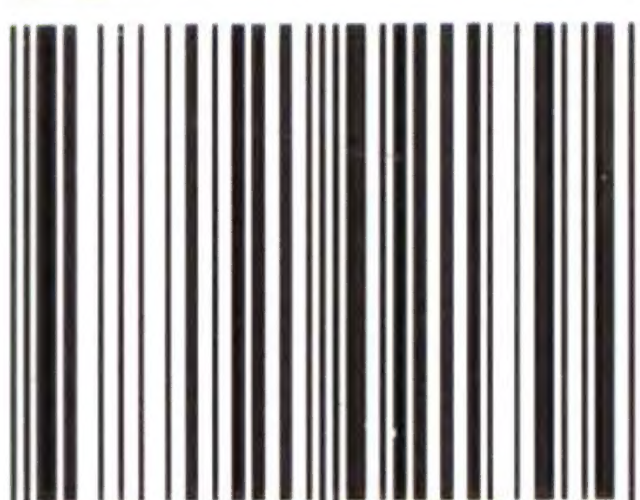
——Donald Yee，Mesa 社会学院

作者简介

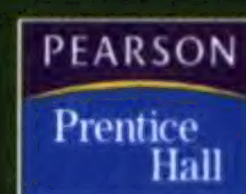
Edward B. Magrab 博士是马里兰大学机械工程系教授，一直从事振动学、声学和设计与制造集成化方面的研究。Magrab 博士与相关领域的专家共同执笔完成了本书。













责任编辑：史　平
责任美编：毛惠庚

ISBN 7-121-02179-X　定价：59.00 元

国外电子与通信教材系列

MATLAB 原理与工程应用

（第二版）

An Engineer's Guide to MATLAB

with Applications from Mechanical, Aerospace, Electrical, and Civil Engineering

Second Edition

[美] Edward B. Magrab　Shapour Azarm
Balakumar Balachandran　James H. Duncan
Keith E. Herold　Gregory C. Walsh　著

高会生　李新叶　胡智奇　等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书的主旨是让读者熟练掌握MATLAB知识，灵活运用到求解广泛的工程问题中去。书中以MATLAB 7.15为基础，介绍MATLAB基本原理及其在各个工程领域中的应用。在原理部分，涉及MATLAB的基本使用方法、MATLAB中矩阵和向量的表示方法及其运算、数据的输入/输出、MATLAB的函数、程序流程控制，以及二维和三维图形的制作。在工程应用方面，介绍了在机械零件设计、动力学与振动、控制系统、流体力学、热传导、优化和工程统计等领域中MATLAB的应用。书中给出了大量翔实的工程应用实例，每章后面附有习题，供读者练习。

本书体系完整，内容丰富，实用性强。既可以作为工科院校大学生和研究生的教学用书，也可以作为工程技术人员利用数值方法求解多种工程问题的参考书。

图书在版编目（CIP）数据

MATLAB原理与工程应用（第二版）/（美）马格雷伯（Magrab, E. B.）等著；高会生等译.
北京：电子工业出版社，2006.1
（国外电子与通信教材系列）
书名原文：An Engineer's Guide to MATLAB: with Applications from Mechanical, Aerospace, Electrical, and Civil Engineering, Second Edition
ISBN 7-121-02179-X

Ⅰ. M...　Ⅱ. ①马... ②高...　Ⅲ. 计算机辅助计算－软件包，MATLAB 7.0－教材　Ⅳ. TP391.75

中国版本图书馆CIP数据核字（2005）第157045号

责任编辑：史　平
印　　刷：北京天竺颖华印刷厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 1092　1/16　　印张：38　　字数：950千字
印　　次：2006年1月第1次印刷
定　　价：59.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个"国外电子与通信教材系列"项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝"国外电子与通信教材系列"项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿
中国工程院院士、清华大学教授
"国外电子与通信教材系列"出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套“国外计算机科学教材系列”，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于“十五”期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进“国外电子与通信教材系列”，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，“国外电子与通信教材系列”的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过“教育部高等学校电子信息科学与工程类专业教学指导委员会”的审核，并得到教育部高等教育司的批准，纳入了“教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书”。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、南京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学、中山大学、哈尔滨工业大学、西南交通大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

译 者 序

MATLAB作为一种数学工具，广泛用于各个工程领域。工程技术人员通过MATLAB提供的工具箱和丰富的调用函数，可以完成复杂工程问题的数值求解，并能够将计算结果形象直观地显示出来。随着应用的不断深入，MATLAB也处在不断发展和完善的过程中。为满足广大读者的需要，马里兰大学Park学院机械工程系Edward B. Magrab教授主持编写了《MATLAB原理与工程应用》(第二版)，由 Prentice Hall 出版发行。本书在理论教学与工程应用方面普遍得到好评。鉴于此，我们进行了翻译工作。

本书在内容组织方面保持了第一版原有的风格，在选材上既不是过分强调MATLAB本身的原理，也不是详细介绍各学科的基础理论，而是重点说明如何用MATLAB所提供的功能解决工程上的实际问题。本书的前7章简要介绍了MATLAB的基本概念及其使用方法，后7章重点介绍MATLAB在机械零件设计、动力学与振动、控制系统、流体力学、热传导、优化和工程统计等方面的应用。本书的前后两部分保持相对独立，内容组织合理，语言叙述精练，工程指导性强。

本书结合MATLAB的发展现状，对原理介绍部分进行了较大调整，以MATLAB 7.15为基础，重点介绍如何建立和使用MATLAB环境，以及不同的使用方法。为了使内容编排更加清晰合理，对各章节采用的素材进行了重新整理，补充了命令一览表和举例列表。对于各章习题的顺序也进行了合理调整，提高了全书的逻辑性和严谨性。为了突出新版MATLAB的功能，本书增加了使用Symbolic Toolbox进行符号数学运算，以及使用System Identification Toolbox进行系统标识等内容。在工程应用方面，扩展了空气动力学、明槽流和两点边界值求解等新的MATLAB工程应用领域。另外，在动力学与振动、流体力学、优化和控制系统等章节中补充了大量实例，对相关内容进行了丰富和完善。

本书由华北电力大学高会生、李新叶、胡智奇、赵振兵、徐扬等多名教师翻译。华北电力大学电子与通信工程系的部分老师和同学也参加了翻译和校对工作。

由于本书涉及的学科范围广，翻译难度大，加之译者水平有限，书中难免有疏漏和错误，欢迎广大读者批评指正。

第二版序言

本书第二版的主要目的仍然是让读者熟练掌握 MATLAB 的知识,并用来求解工程问题。正如本书绪论中所述,近年来,MATLAB 一直在发展。与第一版相比,第二版所做的修改和完善的内容充分反映了 MATLAB 7.14 的新功能。在本书给出的例子中这一点体现得非常明显,很多程序由此变得简单了。

本书第 1 章的内容有较大变化,重点介绍了如何建立和使用 MATLAB 环境,以及不同的使用方法。为了使内容更加清晰,本章的素材进行了重新组织。除此之外,本书还为读者提供了以下便利:在每一章中给出所使用的 MATLAB 命令一览表;全书给出举例列表,以便读者快速查找自己感兴趣领域的代表性实例;在第 1 章中用表格详细说明了命令族所产生的不同结果;统一的印刷字符和字体格式使得计算机代码更具有可读性。

本书增加了与以下新主题相关的素材:使用 Symbolic Toolbox 进行符号数学运算;使用 System Identification Toolbox 进行系统标识;以及 MATLAB 在空气动力学、明槽流和两点边界值求解问题中的应用,等等。另外,在动力学与振动、流体力学、优化和控制系统等章节中补充增加了大量实例和描述。

我校的 George E. Dieter 博士、Emeritus 教授和 Donald Barker 教授对本书第一版提出了重要的反馈意见。Dieter 博士认真阅读了第 1 章至第 8 章和第 14 章的内容,向我们提出了大量改进意见。Barker 教授从事 MATLAB 课程教学多年,积累了丰富的经验和广泛的素材,在本书中与我们共享。最后,我们衷心地希望读者多提宝贵意见和建议。

E.B.Magrab

S.Azarm

B.Balachandran

J.H.Duncan

K.E.Herold

G.C.Walsh

College Park, MD

目　　录

第1章 绪　　论

本章介绍了 MATLAB 的基本特性及其基本语法。

1.1 概述

MATLAB的名称源自 *Mat*rix *Lab*oratory，是一门计算语言，专门以矩阵的形式处理数据。MATLAB 将计算与可视化集成到一个灵活的计算机环境中，并且提供了大量内置函数，可以在广泛的工程问题中直接利用这些函数来求得数值解。

1.2 MATLAB 环境

MATLAB 启动后显示三个窗口，如图 1.1 所示。左上窗口为工作区间窗口，显示用户定义的变量及其属性列表，变量属性包含变量类型及变量长度。工作区间窗口也可显示为当前目录窗口，显示 MATLAB 所使用的当前目录及该目录下的全部文件名。右侧窗口为 MATLAB 命令窗口。左下窗口为历史命令窗口，显示每个工作周期在命令窗口输入的全部命令。工作周期指 MATLAB 启动至退出的工作时间间隔。在每一条命令前，历史命令窗口都显示一个时间和日期，以标明该命令所记录的时间。这十分有利于查看以往工作周期的操作过程，并且在没有其他记录时，还可重新获取以往输入的命令。除非通过 *Edit* 菜单的 *Clear Command History* 子菜单清除所记录的历史命令，否则将一直保存。工作区间窗口和命令窗口具有类似的清除操作，这两项操作将在下面讨论。

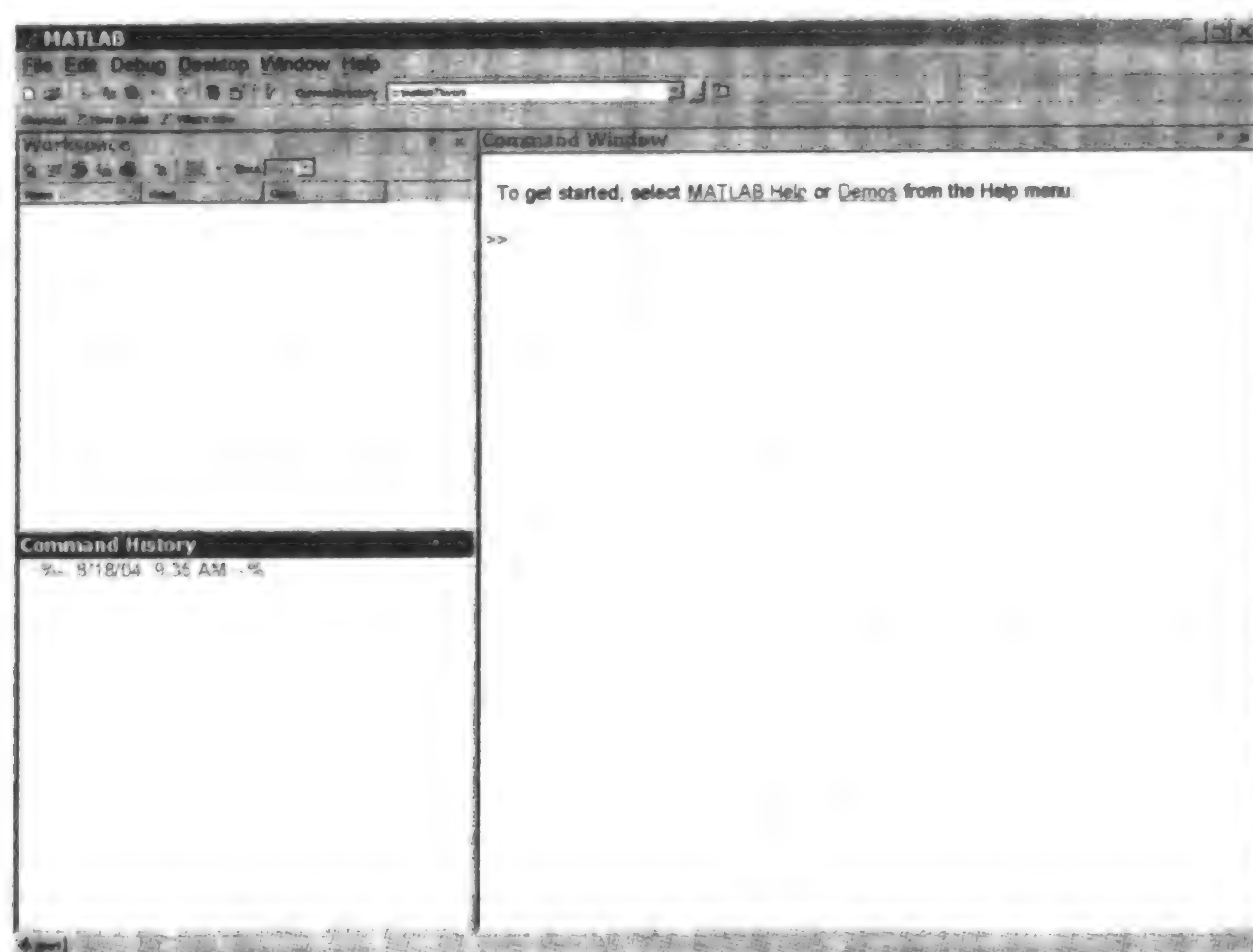


图 1.1　MATLAB 默认窗口

单击 MATLAB 窗口左上角 *File* 菜单下面的白色矩形图标,可打开 MATLAB 编辑/调试窗口,用于创建及运行程序,如图 1.2 所示。其他窗口可通过 *View* 菜单打开并访问。要退出任何窗口,关闭即可。运行 MATLAB 时,可仅使用命令窗口和编辑器窗口,需要时再调用其他窗口,如图 1.3 所示。再次运行 MATLAB 时,系统会自动记录当前配置情况,并按图 1.3 所示显示工作窗口。

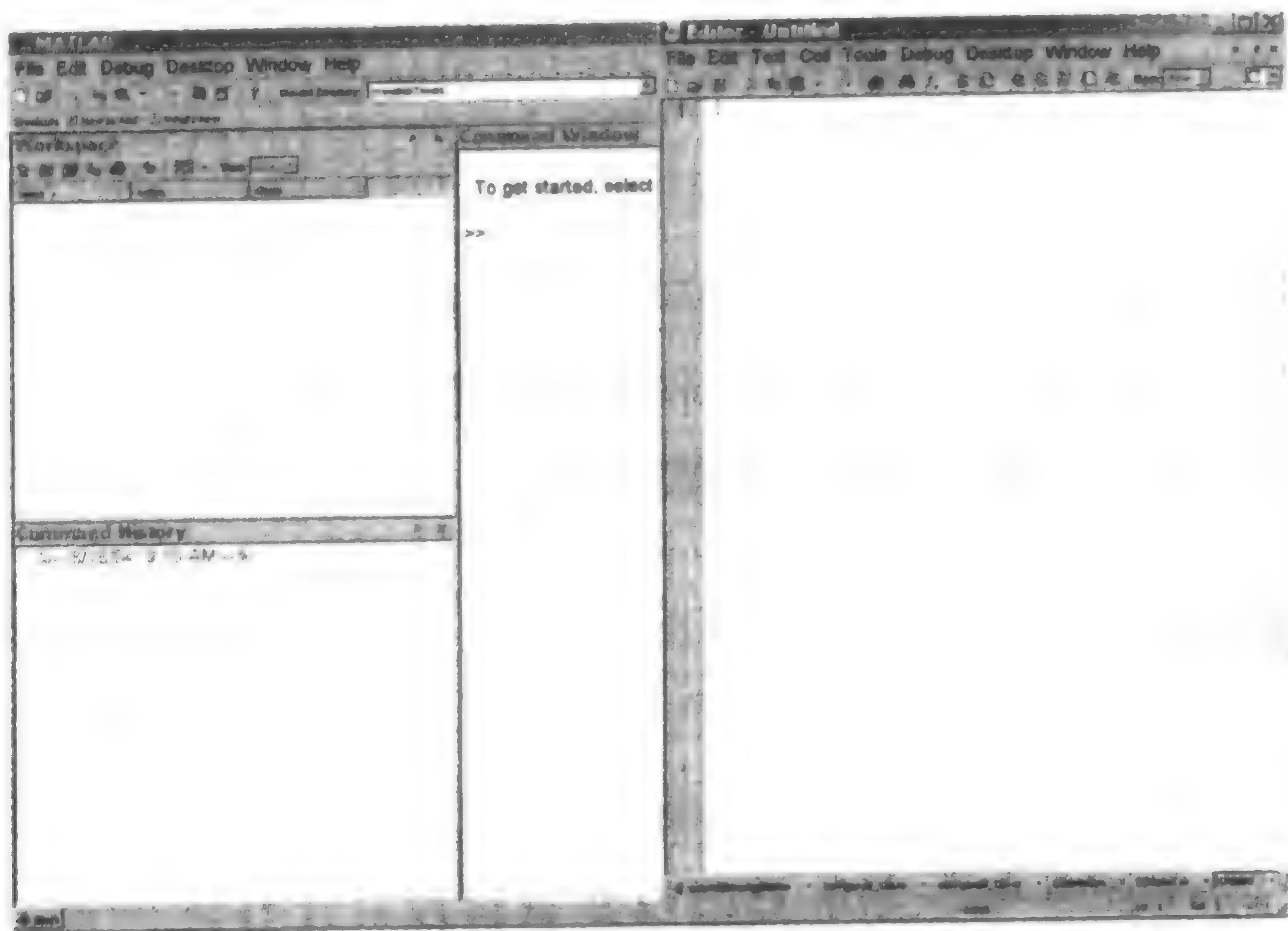


图 1.2　MATLAB 默认窗口和编辑器窗口

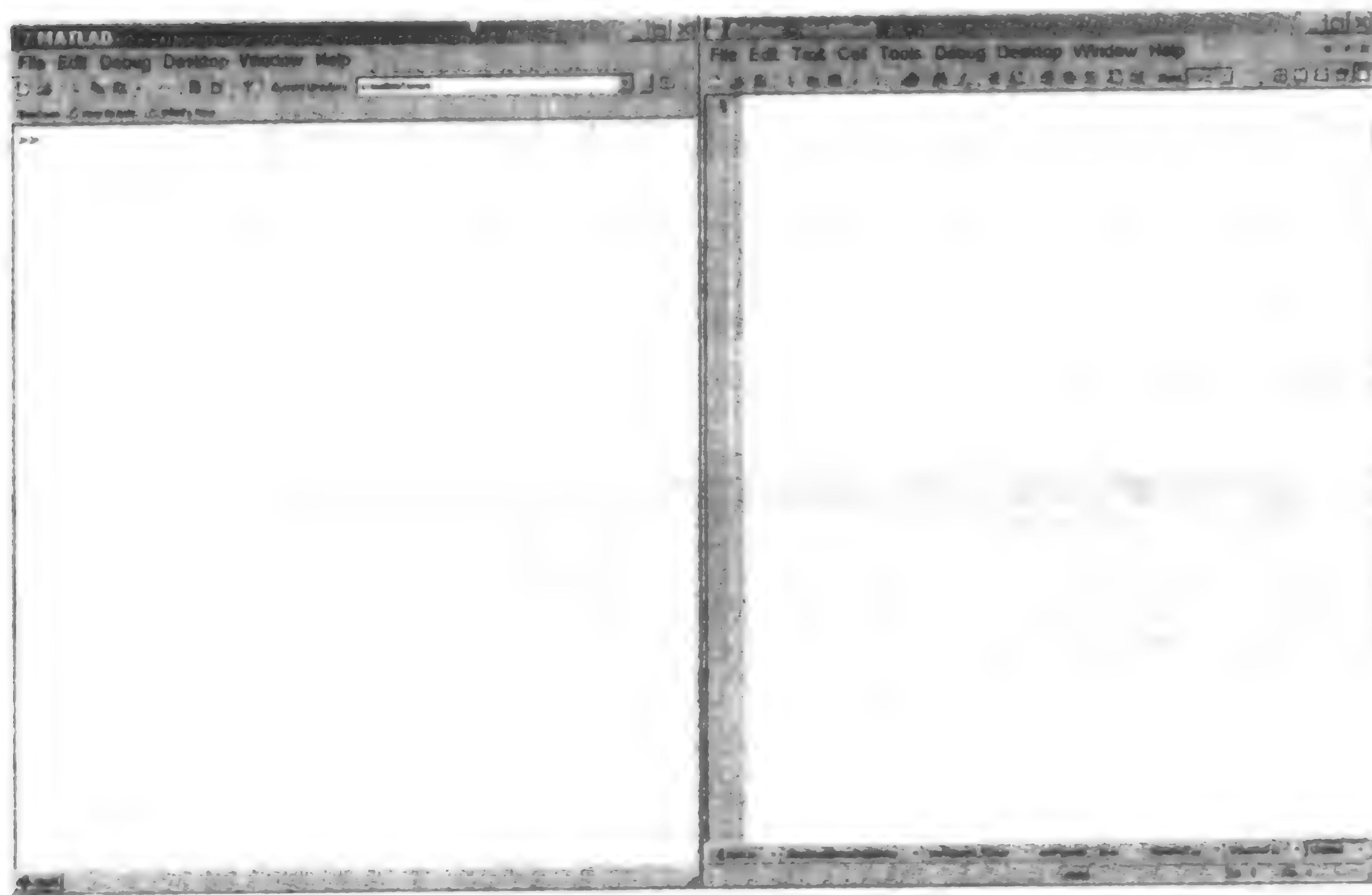


图 1.3　关闭历史命令及工作空间窗口后重新排列的 MATLAB 默认窗口及编辑器窗口

1.3　MATLAB 的使用方法

1.3.1　基础知识——命令窗口管理

在 MATLAB 运行期间(即程序退出之前),除非调用 `clear` 函数,否则 MATLAB 会在内存中保存全部变量值,包括命令行输入的变量以及执行程序文件所引入的变量。清除工作空间变量值也可以通过 *Edit* 下拉菜单中的 *Clear Workspace* 命令实现(参见图 1.4)。`clear` 函数可

清除内存中的所有变量。如前所述,在 MATLAB 运行期间可随时通过输入变量名或从表达式中引用变量名来访问变量的当前数值(在未调用 clear 函数之前)。这些变量可作为全局变量加以引用。

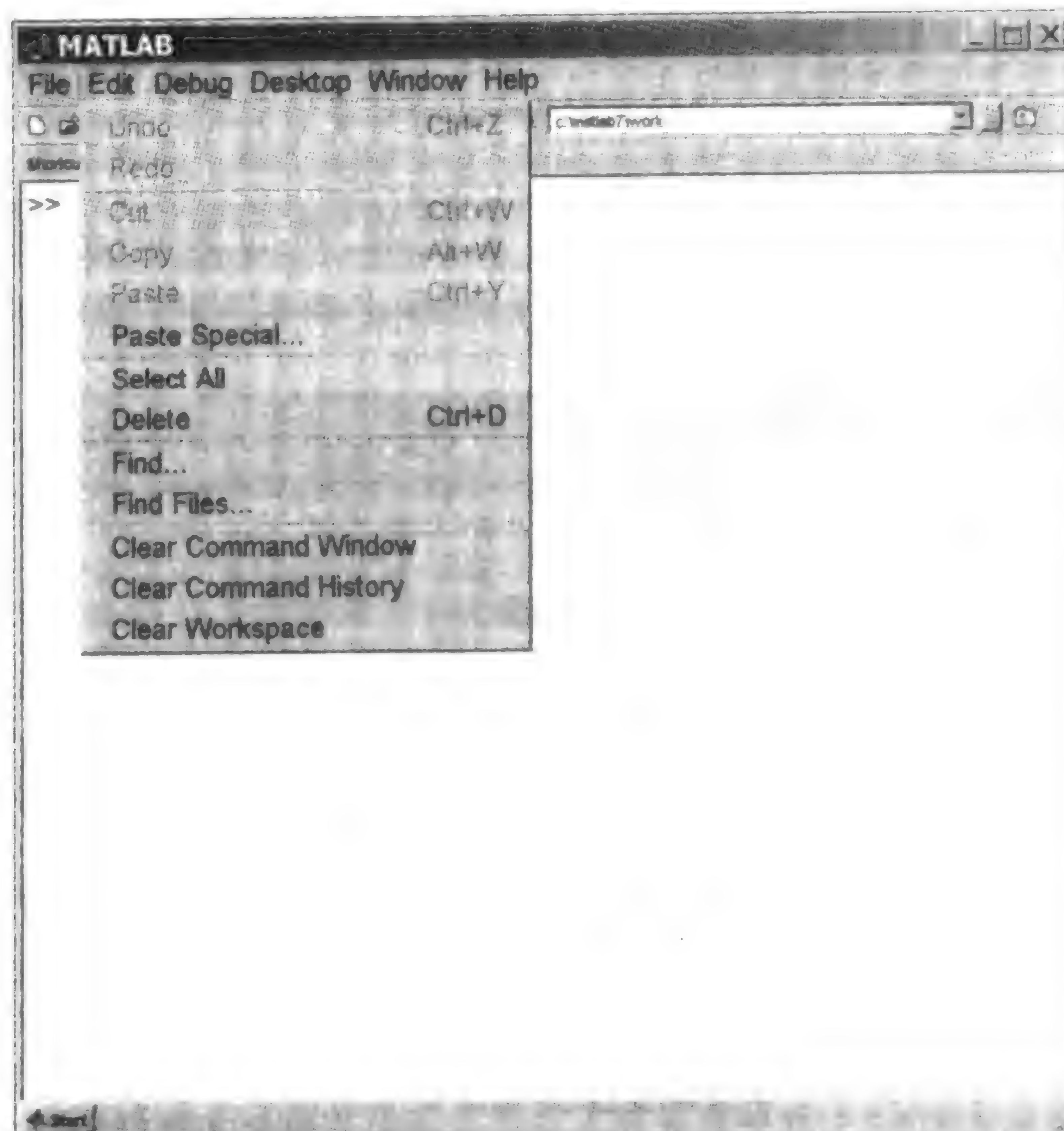


图 1.4 *Edit* 下拉菜单选项

MATLAB 命令窗口输入的信息会保持在窗口中,并可通过滚动条重新访问。一旦信息量超出其滚动内存容量,则最早输入的信息将会丢失。虽然程序文件中所引入的表达式不能重新访问,但程序文件中的变量名和变量值却可重新访问。输入表达式的记录可通过单击 MATLAB 命令窗口顶部 *Edit* 下拉菜单中的 *Clear Command Window* 子菜单加以清除。但应注意,该操作仅清除 MATLAB 的命令窗口内容,而不能删除变量。删除变量只能通过 clear 来实现(参见图 1.4)。也可以通过在命令窗口中输入 clc 命令来清除 MATLAB 命令窗口内容。此外,copy 和 paste 图标可用来实现将 MATLAB 命令窗口中以前输入的表达式复制到当前行,或将 MATLAB 表达式由 MATLAB 命令窗口粘贴到文字处理器窗口,反之亦然。

在 MATLAB 命令窗口中输入 whos 命令或选择 *View* 菜单的 *Workspace* 子菜单,可以创建一个变量信息显示窗口,在此窗口中显示变量列表。这些变量是自上次调用 clear 函数后产生的。显示内容为变量名、变量长度、每个变量所占用的存储字节数及变量类型。变量类型包括:数值型(参见第 2 章)、字符串型(参见 3.1 节)、符号型(参见 5.6 节)、单元型(参见 3.4 节)及 inline 对象(参见 5.2.3 节)等。可通过单击右上角临近 × 的图标改变工作空间窗口位置,最小化工作空间窗口有利于下一次更为便捷地获得变量信息。

为在命令窗口中更加清晰地显示出字母及数字,MATLAB 提供了 format 函数的几种功能选项。其中最为常用的是:

```
format compact
```

和

```
format long e
```

前者剔除了显示中多余的空行或空格,后者可将系统默认的5位数字的显示格式转换为16位数字位加3位指数位的显示格式。调试程序时,在产生的数值变化量很小或者数值变化范围很大的情况下,采用 format long e 选项尤为重要。要恢复系统默认设置,可输入:

```
format short
```

这些属性值也可通过单击 *File* 菜单的 *Preferences* 子菜单,弹出 *Preferences* 设置窗口后选择 *Command Window* 项进行设置。如图1.5所示,通过显示格式列表选择期望的显示格式。

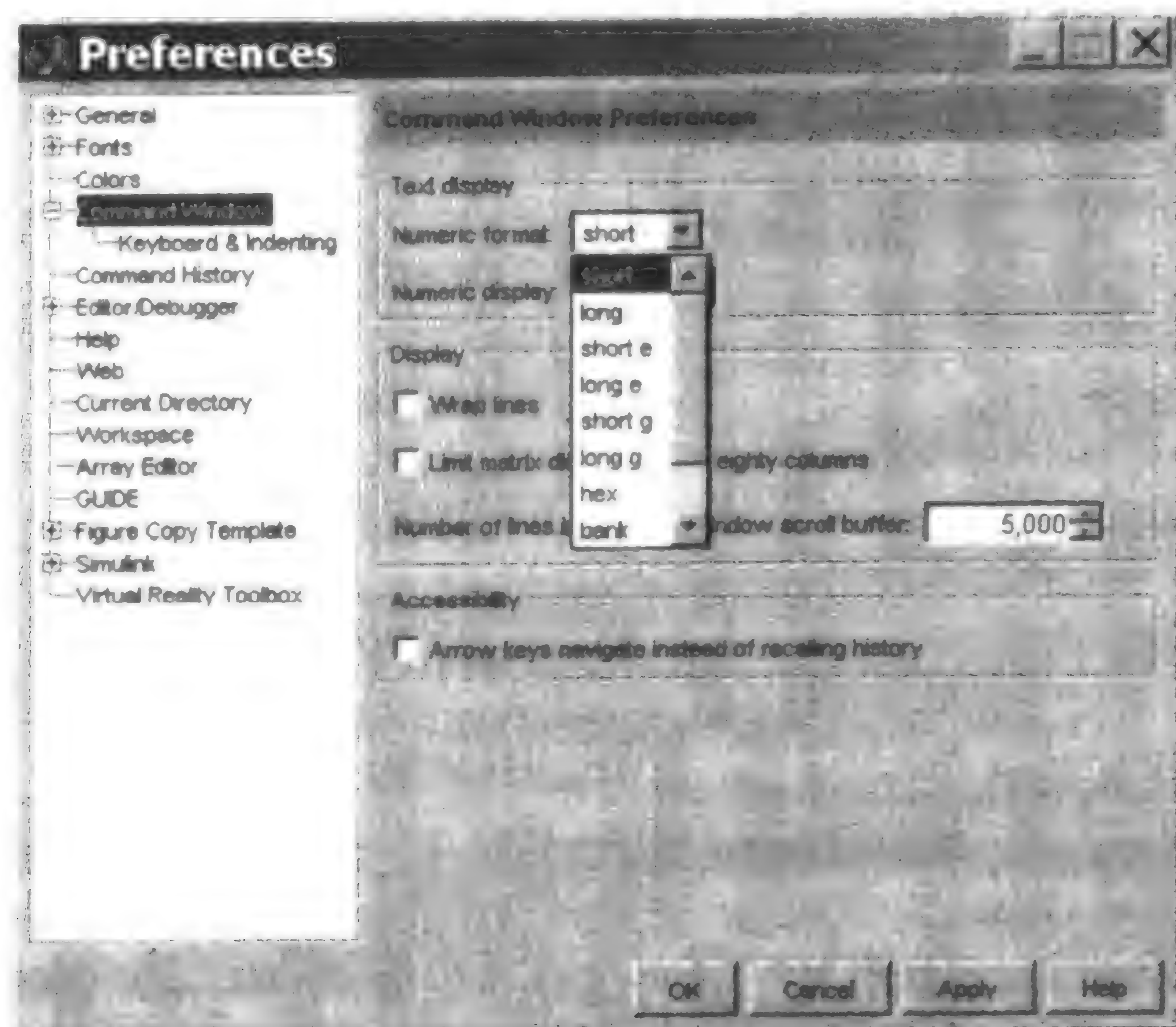


图1.5 *Preferences* 菜单选项

MATLAB 中十分有用的两个组合键为"^c"(即 Ctrl 键与 c 键同时按下)和"^p"(即 Ctrl 键与 p 键同时按下)。"^p"用于将最近一次键入的信息显示在 MATLAB 命令窗口中,按 *Enter* 键可再次执行该命令。按 *Enter* 键执行前,可修改调入的命令行。如果连续两次按下"^p"组合键,可调用再上次键入的命令行,依次类推。也可以通过上、下箭头键完成同样的功能。"^c"用于终止程序或函数的执行,也可用于退出暂停的程序或函数。

1.3.2 在 MATLAB 命令窗口中执行表达式——MATLAB 的基本语法

MATLAB 允许用户创建的变量名不可超过63个字符,多余部分将被忽略掉。变量名要求以大写或小写字母开头,后面可跟大小写字母、数字或下划线。字符间不允许有空格。变量名对于大小写敏感,即变量名 *junk* 与 *junK* 是不同的。MATLAB 中存在两个一般约定:使用下划线和首字母大写。例如,exit pressure 是所要计算的数值,在 MATLAB 命令行、程序或函数中可将其定义为 *exit_pressure* 或 *ExitPressure*。

创建合适的变量名是易于组织的标识符和可读性强的表达式之间的一种折中。如果表达式很长,则较短的变量名将是更为合适的选择。随着符号组合日趋复杂,变量命名变得更为重要。较短的变量名可以减少不恰当的术语和算术符的组合所带来的错误。此外,不允许使用希腊字母,或者上下标字符作为变量名。但是,可以拼写希腊字母或在下标字符前加入下划线。例如,可以将 σ_r 写为 sigma_r,将 c_3 写为 c3 或者 c_3。

MATLAB 在命令窗口运行时要求首先在“>>”提示符后定义一个或多个变量，并要求表达式在使用变量以前一定要先对其进行赋值操作（符号工具箱使用的变量除外）。赋值运算符为等号（=）。对变量赋值时，先输入变量名和等号，然后输入变量数值并按 *Enter* 键结束。如想实现 $p=7.1, x=4.92, k=-1.7$ 功能，必须要在 MATLAB 命令窗口中进行以下的信息交互：

```
>> p=7.1          ←用户输入
p=
  7.1000          ←系统响应
>> x=4.92         ←用户输入
x=
  4.9200          ←系统响应
>> k=-1.7         ←用户输入
k=
 -1.7000          ←系统响应
```

在表达式后加分号（;）可省略系统的响应信息。可按以下形式输入表达式，行尾以按下 *Enter* 键结束。

```
>> p=7.1;
>> x=4.92;
>> k=-1.7;
>>
```

MATLAB 允许在一行中输入多个表达式，行尾以 *Enter* 键结束。表达式间以逗号或分号进行分隔。使用逗号分隔表达式时系统会回显输入的数值。按如下格式输入信息：

```
>> p=7.1, x=4.92, k=-1.7
```

系统显示为：

```
p=
  7.1000
x=
  4.9200
k=
  -1.7000
>>
```

如果用分号分隔表达式，系统不会输出响应信息。

标量的加、减、乘、除和幂运算有 5 个运算符号，分别用 +，-，*，/ 和^ 表示。例如，数学表达式：

$$t=\left(\frac{1}{1+px}\right)^k$$

在 MATLAB 中可写为：

```
t=(1/(1+p*x))^k
```

在执行本语句之前首先要对 p，x 及 k 赋值。否则，会显示错误信息。假定 p，x 及 k 值为前面输入的数值且尚未被清除，则系统显示计算结果：

```
t=
  440.8779
```

在变量 t 的 MATLAB 表达式中使用了圆括号,用来确定运算次序。MATLAB 本身包含默认的运算次序,故可最大限度地减少圆括号的使用数目。为了使表达式更易理解,可增加一些额外的圆括号。默认的运算次序为:幂运算为最高,其次为乘除,最后为加减。无论是在成对的圆括号内,还是在通常的表达式中,MATLAB 都遵循从左到右的计算次序。

表 1.1 MATLA 语法举例

数学表达式	MATLAB 表达式
$1-dc^{x+2}$	1 - d * c^(x+2)
$dc^{x}+2$	d * c^x + 2 或 2 + d * c^x
$(2/d)c^{x+2}$	(2/d) * c^(x+2) 或 2/d * c^(x+2) 或 2 * c^(x+2)/d
$(dc^{x}+2)/g^{2.7}$	(d * c^x + 2)/g^2.7
$\sqrt{dc^{x}+2}$	(d * c^x + 2)^0.5 或 sqrt(d * c^x + 2)

参见表 1.1 中的示例,其中包含了标量数值 c,d,g 和 x。MATLAB 函数

```
sqrt
```

用于求解输入变量的平方根。注意,表 1.1 的第一行,$x+2$ 两边的圆括号是必需的。如果没有圆括号,MATLAB 表达式将变为:

```
1-d * c^x + 2
```

相应的数学表达式则变为 $1-dc^{x}+2$。第三行中的幂运算表达式同样如此,MATLAB 表达式

```
2 * c^(x + 2)/d
```

依据运算规则是正确的。首先计算中间的圆括号($x+2$),然后依据运算规则进行幂运算,接着按照从左到右的次序进行乘除运算。因为依据运算规则,2,c^{x+2},d 具备相同的运算优先次序:乘和除。

MATLAB 语法的注释与说明

空格

数学表达式中,应用空格一般不会影响计算结果,除非表达式位于阵列之中,即表达式位于方括号([])内。2.3 节将讨论这一问题。要求等号右边的赋值语句或变量名中多余的空格必须用算术运算符、逗号或分号进行分隔,但有两种情况例外。其一,当数值为复数时,如 $z=a+jb$ 或者 $z=a+ib$,其中 $i=j=\sqrt{-1}$。MATLAB 程序代码为:

```
a = 2;b = 3;
z = a + j *b % 或 a +b *j
```

运行结果为:

```
z =
  2.0000 + 3.0000i
```

注意,程序代码中为 j,但系统响应结果为 i,这表明系统将两个量等同处理。此外,应注意到 j 并未进行预定义,因此 MATLAB 已默认其等于 $\sqrt{-1}$。但当 a 和 b 直接以数值形式出现在表达式中时,不需要以算术运算符进行分隔。如程序

```
a=2;b=3;
z=(a+j*b)*(4-7j)
```

运行结果为:

```
z=
  29.0000-2.0000i
```

在这种情况下,j 或 i 与数值之间不能有空格。

另一种例外情况为以指数形式表示数值。例如,$x=4.56\times10^{-2}$ 可以表示为:

```
x=0.0456
```

或

```
x=4.56*10^-2
```

或

```
x=4.56e-2
```

最后一个表达式即为指数形式。注意,整数部分的最后一位与指数符 e 之间没有算术运算符。指数符 e 后跟随的位数不超过 3 位。因此,如果以指数形式表示 x^2,则表达式为:

```
x2=4.56e-2^2
```

运行后在命令窗口显示:

```
x2=
  0.0021
```

如果 x 值为 4.56×10^2,隐含的加号可以省略。x 平方可写为:

```
x=4.56e2^2
```

或

```
x=4.56e+2^2
```

系统变量名分配

命令窗口作为计算器应用且未将计算结果分配给表达式时,MATLAB 默认将计算结果分配给变量名 *ans*。例如,如果希望获得 $\pi/3$ 的余弦值,只要在命令窗口[①] 输入以下信息:

```
cos(pi/3)
```

① 在命令窗口中,字符和数字以相同的字体显示。为更好地理解表达式,本节结尾将讨论利用不同字体进行显示的方法。

系统显示为：

```
ans =
  0.5000
```

现在，变量 *ans* 可以像其他变量一样进行应用。如果想在上面结果的基础上加 2，则可以在命令窗口中输入：

```
ans + 2
```

系统显示为：

```
ans =
  2.5000
```

变量 *ans* 被赋予新的数值 2.5。此前赋值为 0.5 的变量 *ans* 不可再访问。

标量与数组

MATLAB 将所有变量视为数组。因此，使用五个算术运算符(+ ， - ， * ， /和^)时必须遵守线性代数规则。这些规则将在 2.6 节进行讨论。变量为标量数值(即为一行一列的单元素数组)时适用代数规则，变量为数组时则代之以点积规则。点积规则将在 2.5 节进行讨论。

复数值

MATLAB 允许进行实数和复数的复合运算，不必采取其他特别措施。因此，如果在命令窗口输入以下信息：

```
z = 4 + sqrt( - 4)
```

则系统显示为：

```
z =
  4.0000 + 2.0000i
```

MATLAB 包含了大量的初等函数。表 1.2、表 1.3 和表 1.4 列出了一些初等函数和一些内置常量。这些函数的参数可以为标量、向量或矩阵。2.3 节和 2.4 节详细介绍了 MATLAB 中向量和矩阵的创建和定义。

表 1.2 部分 MATLAB 初等函数

数学函数	MATLAB 表示
e^x	exp(x)
$\sqrt{x}$	sqrt(x)
$\ln(x)$ 或 $\log_e(x)$	log(x)
$\lg(x)$	log10(x)
$\|x\|$	abs(x)
符号函数(x)	sign(x)

表 1.3　MATLAB 三角函数和双曲函数

	三角函数		双曲函数	
数学函数	原函数	反函数	原函数	反函数
正弦	sin(x)	asin(x)	sinh(x)	asinh(x)
余弦	cos(x)	acos(x)	cosh(x)	acosh(x)
正切	tan(x)	atan(x)①	tanh(x)	atanh(x)
正割	sec(x)	asec(x)	sech(x)	asech(x)
余割	csc(x)	acsc(x)	csch(x)	acsch(x)
余切	cot(x)	acot(x)	coth(x)	acoth(x)

① atan2(y,x)为四象限形式。

表 1.4　部分 MATLAB 常量

数学常量	MATLAB 表示
π	pi
$\sqrt{-1}$	i 或 j(用来表示复数 a+jb,其中 a 和 b 为实数)
浮点相对精度	eps——从 1.0 到下一个最大浮点数的距离
∞	inf

作为内置函数的应用实例,当 $x=1.0, a=0.5$ 时计算下列表达式的值:

$$y=\sqrt{\left|e^{(-\pi x)}-\sin(x)/\cosh(a)-\ln_e(x+a)\right|}$$

可通过下列程序计算表达式的值:

```
x=0.1;a=0.5;
y=sqrt(abs(exp(-pi*x)-sin(x)/cosh(a)-log(x+a)))
```

其中,pi $=\pi$。运行结果如下:

```
y=
   1.0736
```

MATLAB 中变量名的选择没有限制,但应尽量避免设定与 MATLAB 内置函数名或用户自定义函数名相同的变量名。MATLAB 允许内置函数名进行重载。例如,下列表达式即为 MATLAB 合法的表达式:

```
cos=a+b*x^2;
```

其中,cos 与内置余弦函数 cos(x)的函数名相同。建议应尽量避免这种变量名的重复定义,除非当变量 i 和 j 重新定义为实数时(一般为整数)。

MATLAB 函数 fix,round,ceil 和 floor 用于将小数按不同的圆整化规则圆整化为与之最接近的整数值。表 1.5 简要描述了这四个函数的应用情况。

表 1.5　MATLAB 小数到整数转换函数

MATLAB 函数	x	y	说明
y = fix(x)	2.7 -1.9 2.49-2.51j	2.0000 -1.0000 2.0000-2.0000i	舍去小数部分
y = round(x)	2.7 -1.9 2.49-2.51j	3.0000 -2.0000 2.0000-3.0000i	四舍五入
y = ceil(x)	2.7 -1.9 2.49-2.51j	3.0000 -1.0000 3.0000-2.0000i	将数变大
y = floor(x)	2.7 -1.9 2.49-2.51j	2.0000 -2.0000 2.0000-3.0000i	将数变小

MATLAB 函数 complex,abs,conj,real 和 imag 用于创建和操作复数值。表 1.6 简要介绍了这五个函数的应用情况。

表 1.6　MATLAB 复数运算函数

MATLAB 函数	z	y	说明
z = complex(a,b)	a+b*j	-	代表复数,a 和 b 是实数
y = abs(z)	3+4j	5	绝对值
y = conj(z)	3+4j	3-4j	复数共值
y = real(z)	3+4j	3	实部
y = imag(z)	3+4j	4	虚部

除五个算术运算符外,MATLAB 还保留了一些具有特殊意义的字符(+,-,*,/和^)。表 1.7 列出了这些字符,第 2 章到第 5 章将对其进行更深入的讨论。

表 1.7　特殊字符及其使用说明[①]

符号	名称	用途
.	句号	(a) 小数点 (b) 向量或矩阵的一种操作类型,称为点操作。例如 c = a.* b
,	逗号	(a) 参数分隔符,矩阵元素分隔,如 b(2,7);函数参数分隔,如 besselj(1,x);创建向量参数分隔,如 v=[1,x];函数参数输出分隔,如[x,s]=max(a) (b) 几个表达式在同一行时放在每个表达式之后
;	分号	(a) 放在表达式末尾不显示计算结果 (b) 在创建矩阵的语句中指示一行的结束,例如 m=[x y z; a b c]
:	冒号	(a) 创建向量的表达式分隔符,如 x=a:b:c (b) 对矩阵 z 而言,写为 z(:,k)时表示所有行;写为 z(k,:)时表示所有列
( )	圆括号	(a) 矩阵 z 中某一个元素的下标指示,如 z(j,k)表示矩阵 j 行 k 列的元素 (b) 算术表达式分隔符,如 a^(b+c) (c) 函数参数分隔符,如 sin(x)
[ ]	方括号	创建一组数值、向量、矩阵或字符串(字母型)

(续表)

符号	名称	用途
{ }	大括号	创建单元矩阵或结构
%	百分号	注释分隔符。用于指示注释的开始,MATLAB 编译器会忽略其右边的内容。但用于一对引号内部定义字符串时除外,如 a = 'pl = 14% of the total'
'	引号或撇号	(a) '*Expression*' 表明 *Expression* 为字符串(字母型) (b) 表示向量或矩阵的转置
...	省略号	表示 MATLAB 表达式继续到下一行,增强代码的可读性
	空格	上下文相关:或者忽略而作为数据创建语句的分隔符,如 c = [a b];或者作为字符串语句的一个字符
@	At 符	在函数名前放置@创建函数句柄,如@FunctionName
\	反斜线	矩阵特定操作算术运算符

① 也可参见表 4.1。

程序符号约定

为便于识别变量名和数值,程序的输入/输出应遵循以下约定:

变量名	字体	示例
用户创建变量	Times Roman	ExitPressure, a2, sig
MATLAB 函数	Courier	`cosh(x), pi`
用户创建函数	Times Roman Bold	**BeamRoots**(a,x,k)
数值	**字体**	**示例**
程序提供	Times Roman	5.672
输出到命令窗口或图形	Helvetica	5.672

1.3.3 在 MATLAB 编辑器中创建并运行程序

在命令窗口中输入多个表达式或重复输入一组表达式,将会使用户感到很麻烦。为避免这一问题,MATLAB 定义了程序文件:包含一组命令的文件,文件中每一命令行的执行都同在命令窗口里的执行一样。程序文件可由文字处理器、文本编辑器或由 MATLAB 提供的文本编辑器及调试器创建,以文本文件形式保存,扩展名为".m",通常称为 M 文件。如果程序文件是由文字处理器或文本编辑器创建的,在 MATLAB 命令窗口中仅输入不带扩展名的文件名即可执行。如果程序文件是由 MATLAB 编辑器创建的,可以采用上述方法或单击编辑器窗口顶部的 *Save and Run* 图标执行程序文件,如图 1.6 所示。程序文件执行前首先要求保存,第一次保存文件时可单击 *File* 菜单的 *Save As* 子菜单。文件名命名约定如同变量名命名约定一样:以大写或小写字母开头,后面跟随不超过 62 个连续的字母、数字或下划线符号。文件名不允许有空格(这不同于 Windows 操作系统中文件的命名约定)。使用 MATLAB 编辑器生成程序文件时,".m"后缀将自动添加到文件名末尾。应用其他工具生成程序文件时,需要在创建程序文件时手动添加".m"后缀。

由 MATLAB 编辑器创建的另一种文件形式为函数文件,该文件不能在命令窗口中创建。创建函数文件的原因有两个:一个是 MATLAB 内置函数的需要,另一个则是用户希望实现更好的程序管理任务。不同于程序,函数可通过结构化接口实现程序管理。函数也不同于以命令

行方式输入的表达式。在函数方式下,MATLAB 为其分配专有的工作空间,并且可与 MATLAB 环境形成独立的输入/输出关系。第 5 章将进一步对函数进行讨论。

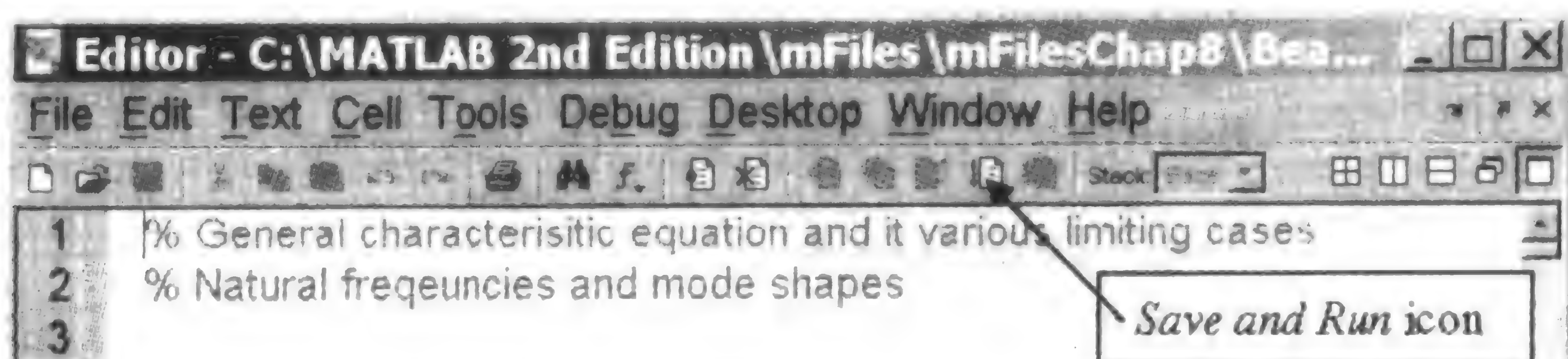


图 1.6　*Editor* 中 *Save and Run* 图标

MATLAB 编辑器具备几个特性,使其非常适合于创建程序文件和函数文件。编写程序代码时,可以方便地将一行或几行程序代码转换为注释语句,或将注释语句转换为程序执行语句。转换时,首先用光标将转换行高亮显示,然后选择窗口顶部 *Text* 下拉菜单中适合的相应操作即可(如图 1.7 所示)。采用第 4 章描述的程序流程控制结构时,可将程序代码缩进以提高程序的可读性。代码缩进时首先用光标将代码行高亮显示,然后按下"^i"组合键(即同时按下 Ctrl 键和 i 键)。单击 *Text* 下拉菜单的 *Smart Indent* 子菜单可实现同样的功能(如图 1.7 所示)。

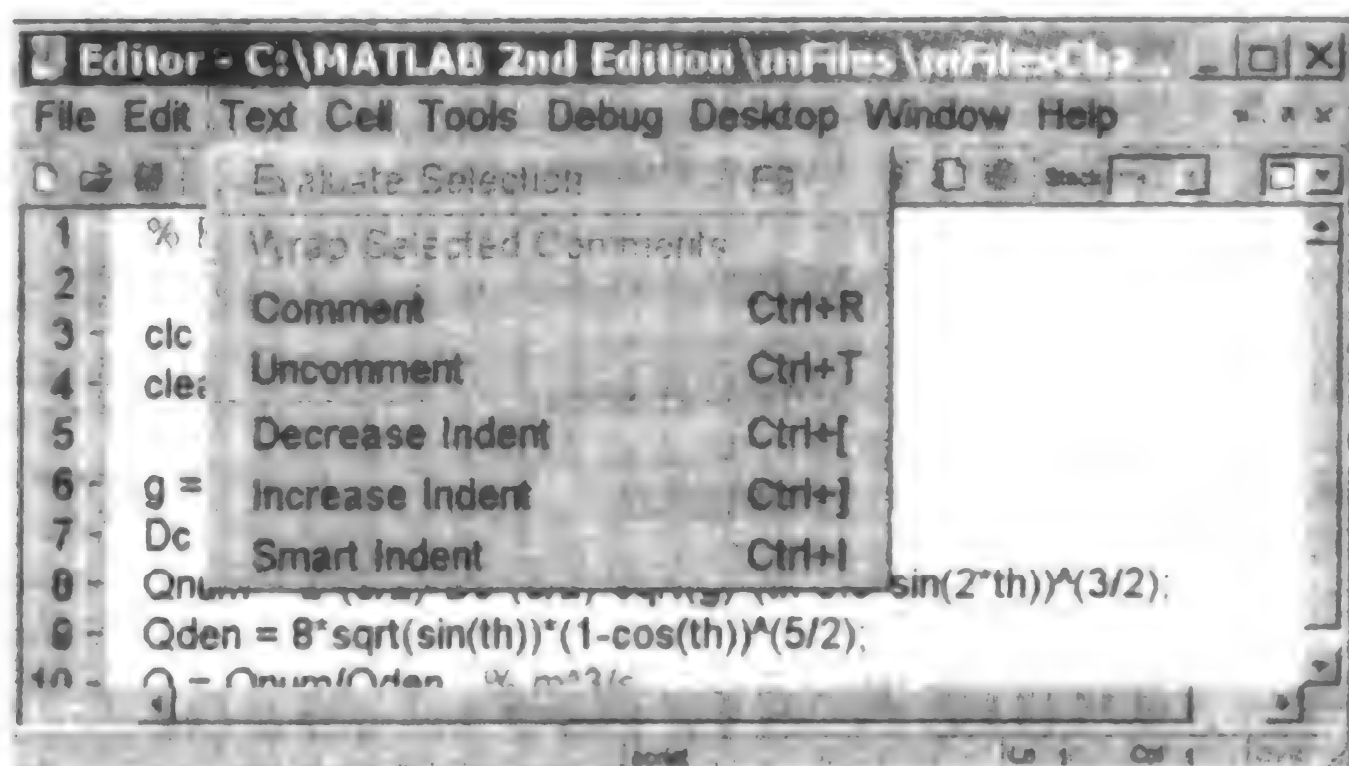


图 1.7　编辑器中 *Text* 的下拉菜单

MATLAB 编辑器具有程序代码显示颜色方案。默认键显示为蓝色,撇号中间的字母和数字显示为红色,注释文本显示为绿色。此外,输入代码行时,编辑器会自动跟踪圆括号对的起止位置。输入右括号时,左侧与之相对应的左括号会瞬间高亮显示。这一特性可实现圆括号对的完整性校验。

本书着重介绍了 MATLAB 命令窗口和编辑器的交互应用。无论命令代码多么少,本书都推荐将其在编辑器中编写为程序代码并运行。

在以下条件下可采用程序文件的形式:

1. 程序包含多行代码
2. 程序将重复使用
3. 希望永久保存
4. 要求随时更新
5. 需要进行大量调试
6. 需要将程序分发给其他人或组织

此外,程序或函数通常具备以下属性:

1. 说明部分,最少应包含以下内容:
   程序用途及所执行的操作
   程序员名
   创建日期
   修改日期
   输入量描述:数值、含义及类型
   输出量描述:数值、含义及类型
2. 输入部分,对输入值进行全面检验,保证所有输入值的正确性,使程序或函数能正确执行。
3. 初始化部分,给变量分配初始数值。
4. 计算部分,完成主要的数值计算。
5. 输出部分,计算结果以图形方式或格式化方式输出。

执行程序文件时,MATLAB 需要提供文件所在的目录路径。设定文件目录路径时,单击 *File* 下拉菜单的 *Set Path* 子菜单,打开 *Set Path* 设置窗口(如图 1.8 所示),单击 *Add Folder* 按钮,在弹出的浏览路径窗口中选定文件所在目录(如图 1.9 所示)。

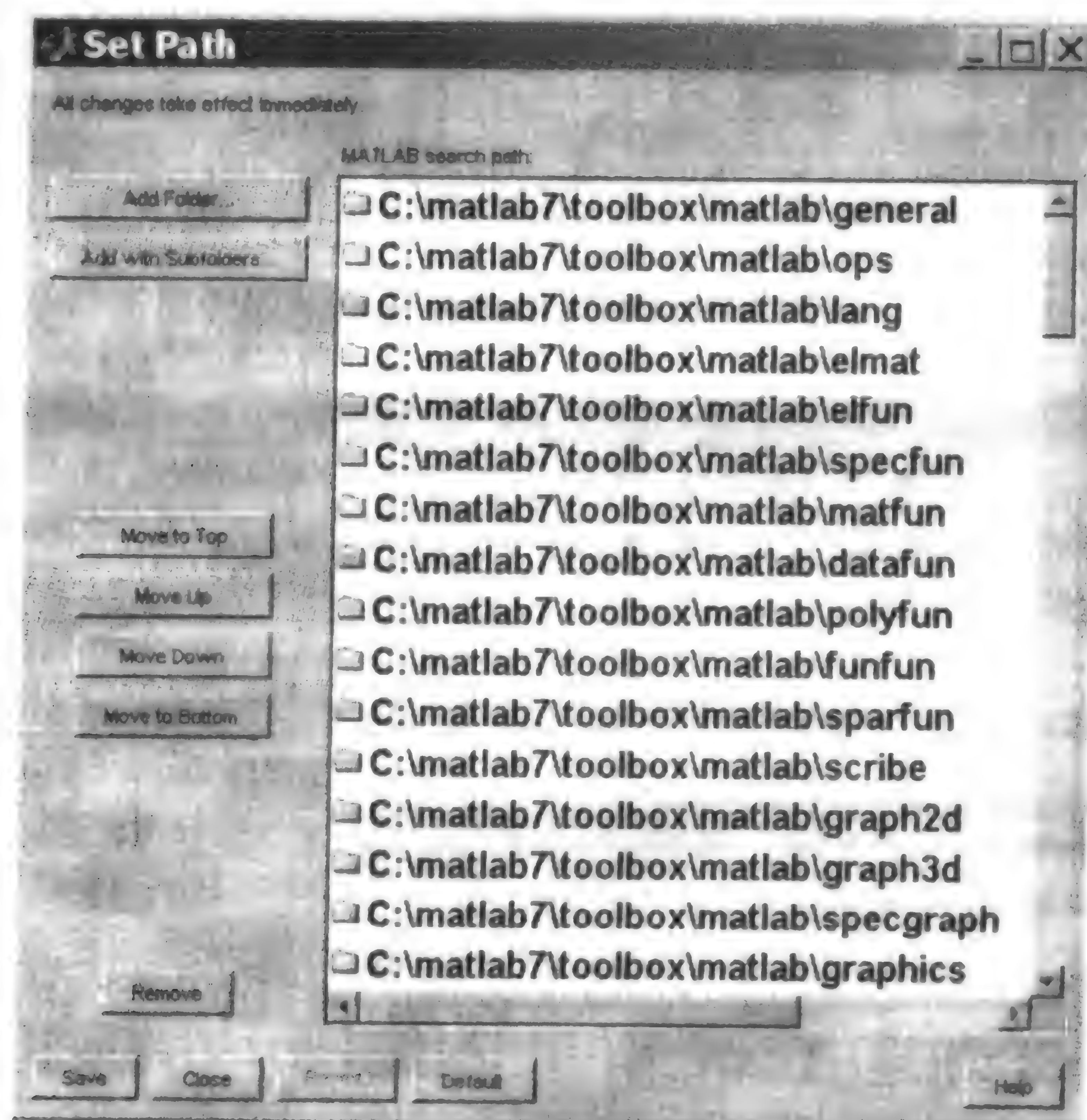


图 1.8 *Set Path* 窗口

退出 *Path Browser* 窗口前,建议单击 *Save* 按钮保存 MATLAB 下一次的应用路径。如果要在编辑器窗口中运行非当前路径程序文件,MATLAB 会弹出如图 1.10 所示的窗口,询问是否将当前路径改变为文件所在路径,确定则单击 *OK* 按钮。此外,也可以通过单击命令窗口中路径设定图标设定当前路径名(如图 1.11 所示)。单击路径设定图标将弹出路径浏览窗口,在浏览窗口中设定当前目录。目录设定后,在编辑器中选择 *File* 菜单的 *Open*, *Save* 或 *Save As* 子菜单时,系统操作目录将自动改变为当前设定目录。

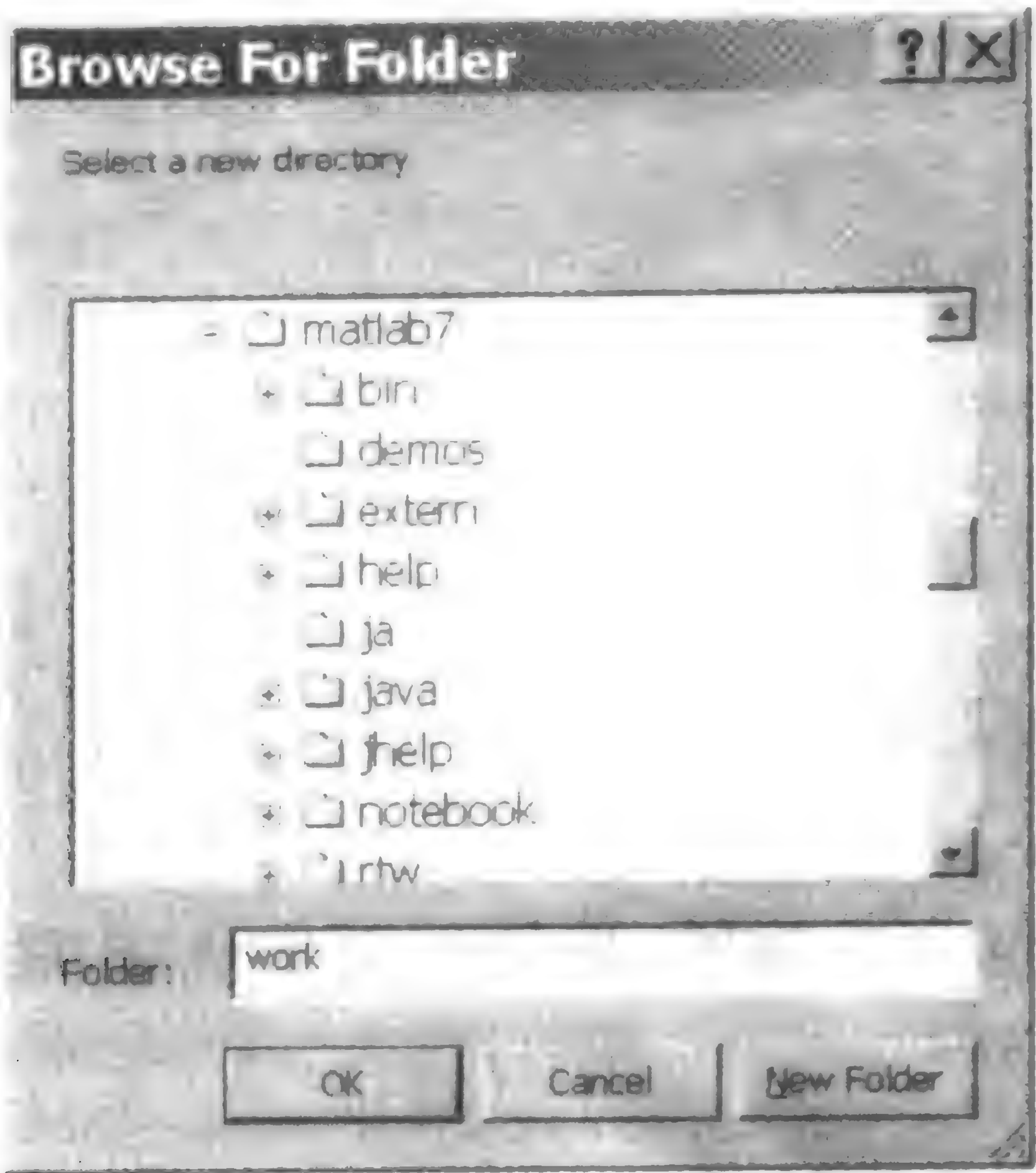


图 1.9 定位目录的弹出窗口

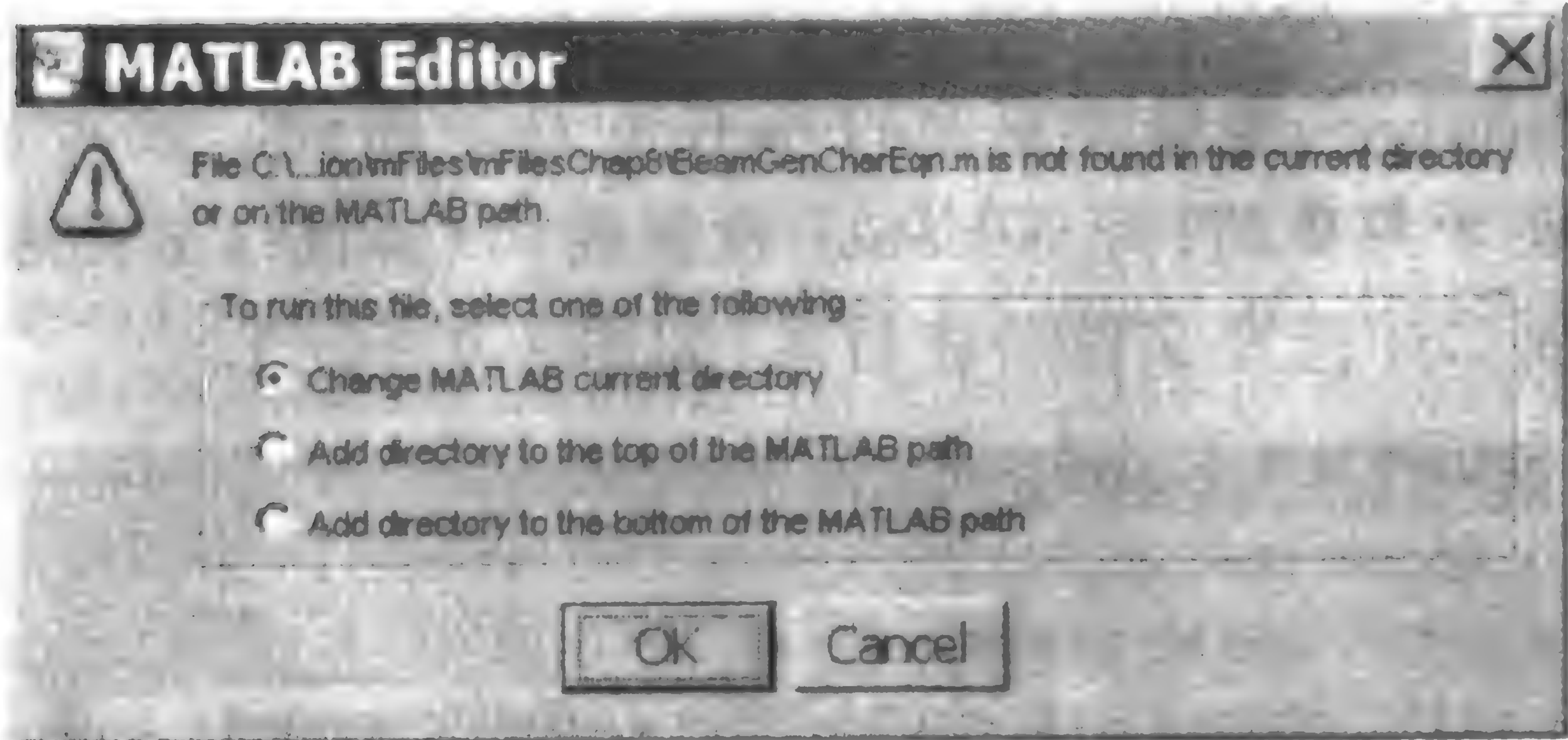


图 1.10 改变当前目录到目标文件目录的弹出窗口

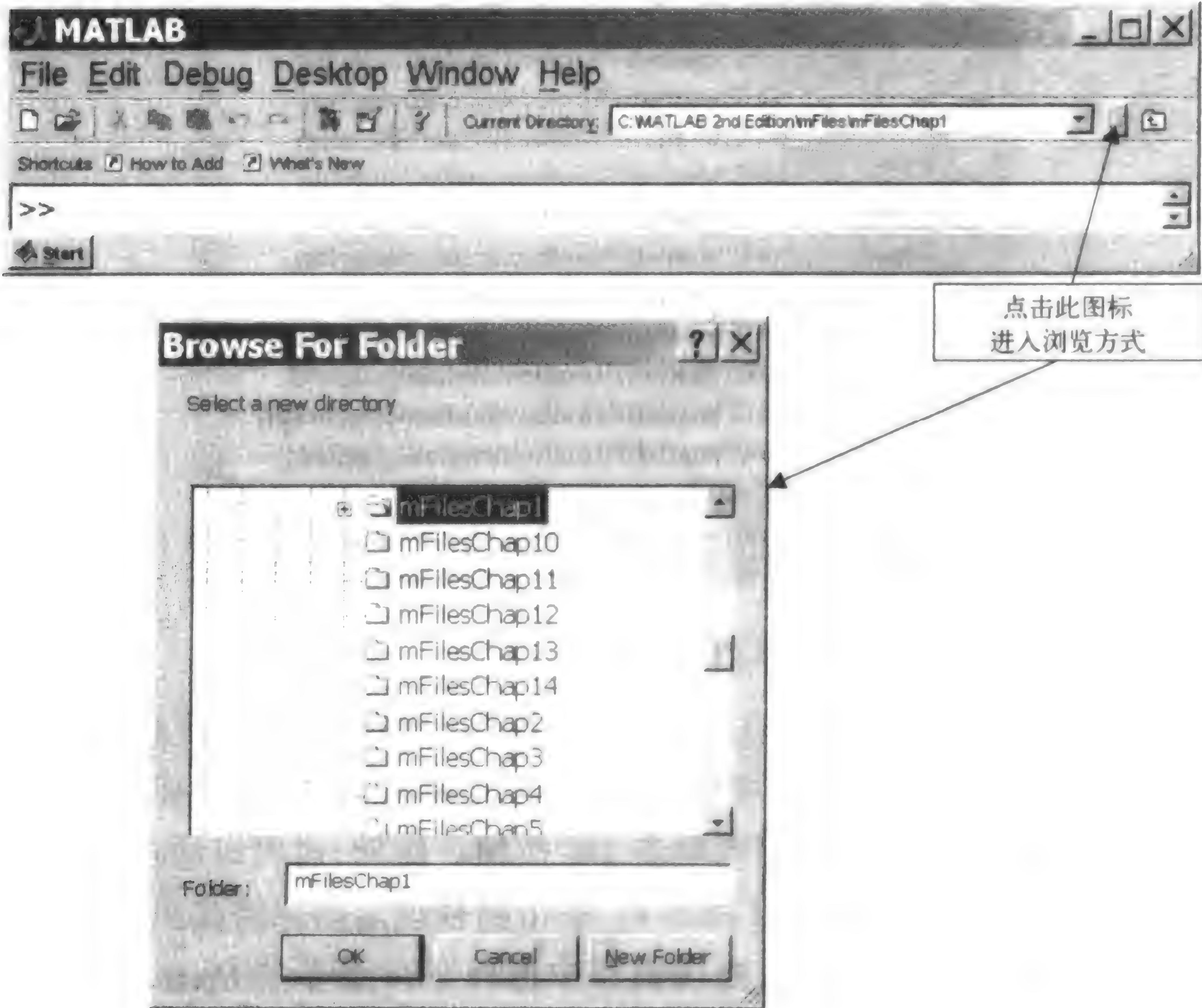


图 1.11 访问 *Browser* 窗口改变当前路径(目录)

程序或函数需要用户在命令窗口中输入数值(向量或矩阵则需要输入一系列数值,参见2.3节和2.4节)时,应包含如下语句:

```
VariableName = input('Any message')
```

其中,input 为 MATLAB 函数,*Any message* 为命令窗口显示信息。语句执行后,显示响应信息,输入数值(或一系列值)后回车,将赋值给变量 *VariableName*。其他数据输入方法将在3.3节进行讨论,在3.2节将进一步说明 input 函数的使用。

有两种方法可实现程序运行结果在命令窗口中的显示。一种方法是在表达式末尾省略分号。此时,MATLAB将在命令窗口显示变量名,其后为等号,然后转到下一行显示变量值。这种方法在调试过程中非常有用。当需要更清晰地显示输出数据时,可采用函数:

```
disp
```

或

```
fprintf
```

3.1节将专门介绍这两个函数的应用。

编写程序时,应在程序代码前加入 clc 和 clear 函数,这样可清除以前定义的所有变量并清空命令窗口。此外,如第6章和第7章所述,在不同窗口中创建图形时,应利用 close all 将窗口全部关闭。如5.2.2节所述,创建全局变量时首先应利用 clear global 清除此前定义的全局变量。函数 clear all 同样可清除已定义的全局变量。因此,一般情况下编写程序代码时首先应输入下列函数:

```
clear % 或 clear all
clc
clear global % 如果已输入 clear all 则不需要输入此函数
close all
```

下面举例说明上述内容,并给出命令输入后的结果显示窗口。

例 1.1 圆形管道中的流体

如图1.12所示,圆形横截面开口管道流体的流量 Q(单位:m^3/s)由下式给出[①]:

$$Q = \frac{2^{3/2} D_c^{5/2} \sqrt{g}(\theta - 0.5\sin(2\theta))^{3/2}}{8\sqrt{\sin\theta}(1-\cos\theta)^{5/2}}$$

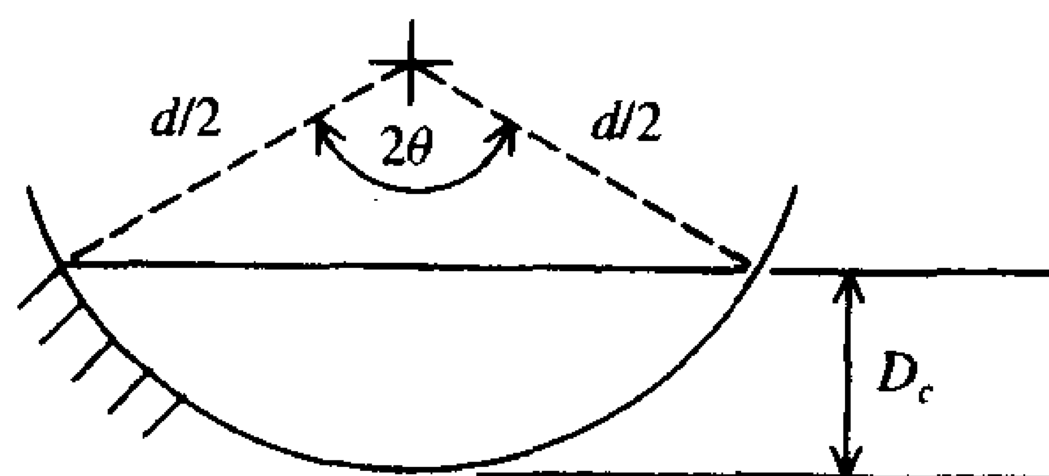


图1.12 圆形管道

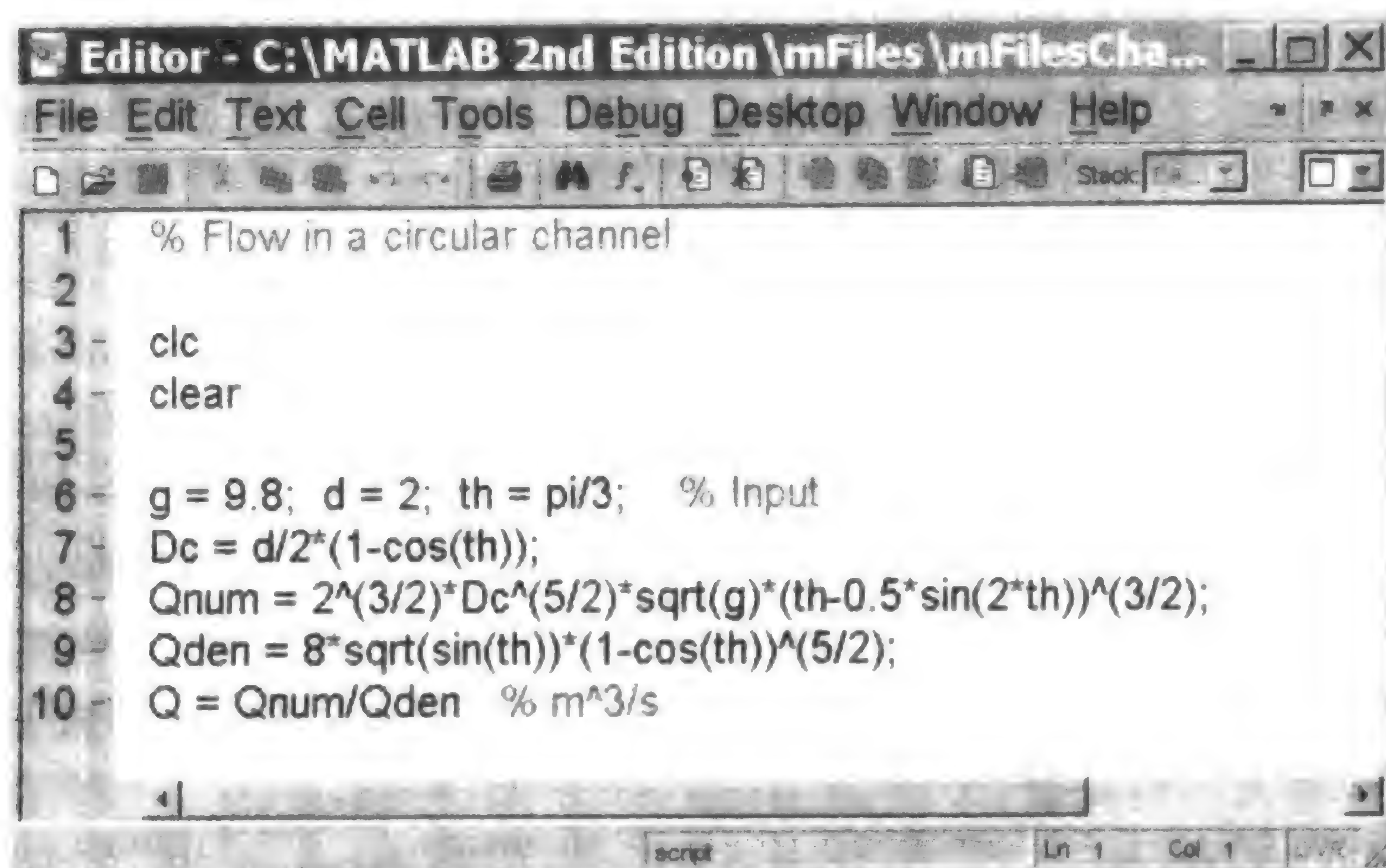
① 参见 T. G. Hicks, *Mechanical Engineering Formulas: Pocket Guide*, McGraw-Hill, 纽约, 2003年,第254页。

其中,$g=9.8\ \mathrm{m/s^2}$ 为重力加速度,D_c 由下式给出:

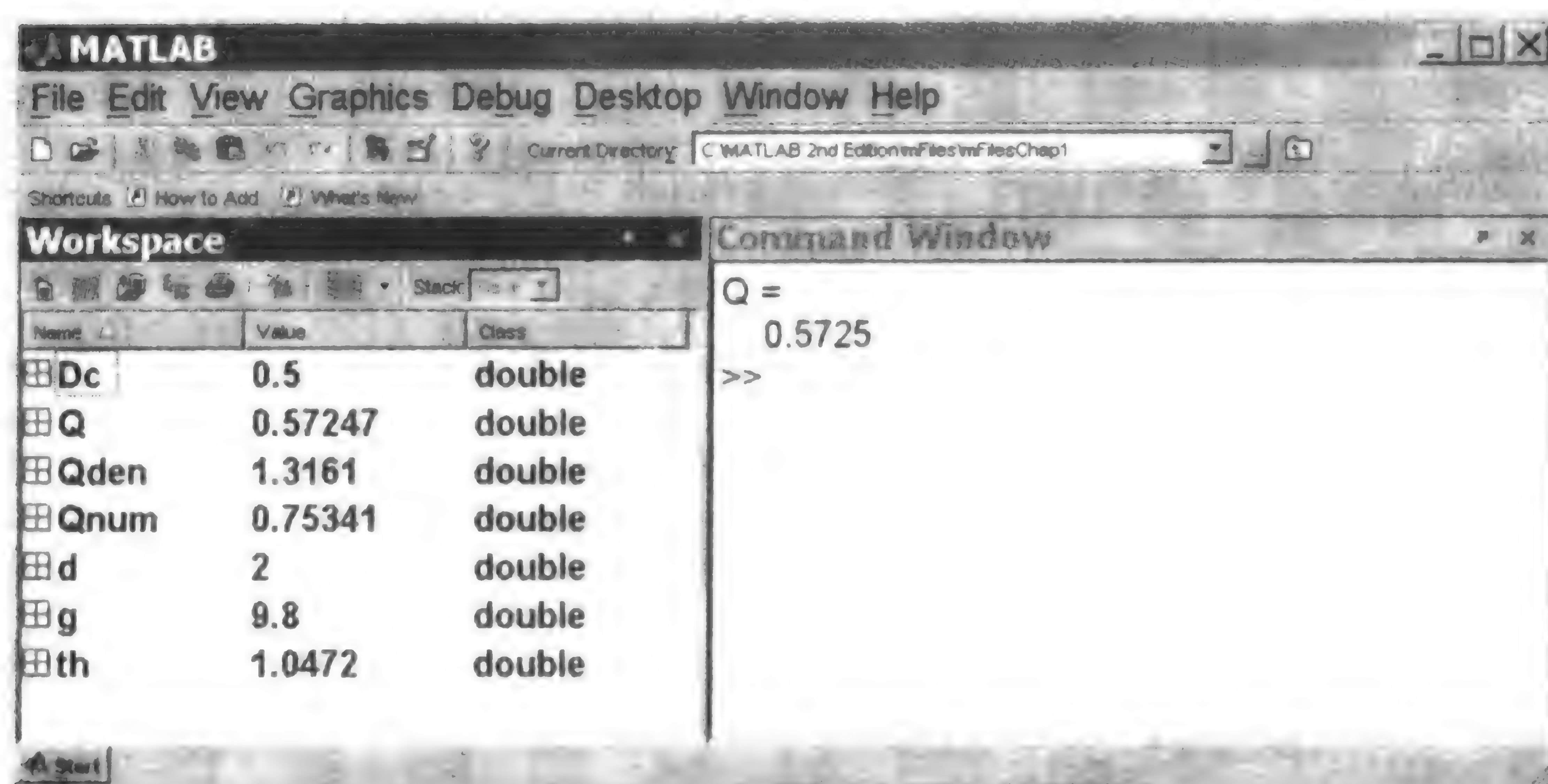
$$D_c = \frac{d}{2}(1-\cos\theta)$$

假设 $d=2\ \mathrm{m}$, $\theta=60°=\pi/3$, MATLAB 程序代码如图 1.13(a)所示。为清楚起见,重复以下过程:

```
g = 9.8; d = 2; th = pi/3; % Input
Dc = d/2 * (1 - cos(th));
Qnum = 2^(3/2) * Dc^(5/2) * sqrt(g) * (th - 0.5 * sin(2 * th))^(3/2);
Qden = 8 * sqrt(sin(th)) * (1 - cos(th))^(5/2);
Q = Qnum/Qden    % m^3/s
```



(a)



(b)

图 1.13 (a) 例 1.1 在编辑器中的程序;(b) 工作空间和命令窗口

单击编辑器 *Save and Run* 图标,在命令窗口显示计算结果 $Q=0.5725$。程序文件执行后,工作空间窗口如图 1.13(b)所示。其中显示了 *Dc*, *Q*, *Qnum*, *Qden*, g, *d* 和 *th* 等已定义的七个变量的相关信息。因为所有命令都在编辑器窗口中实现,历史命令窗口为空,故不加以显示。

最后要说明的是,程序单独定义了各中间变量并分别计算其数值,一则利于程序执行,二则可以实现与程序给出数值的比较。调试过程中,可分别计算各数值,并将其与临时省略各表达式结尾分号所获得的程序计算值进行比较。

1.4 在线帮助

MATLAB拥有完整的在线帮助,可通过几种方法对其进行访问。一种是单击工具栏上的问号图标,打开如图1.14所示的*Help*窗口。为便于引用,每次查询后,应将*Help*窗口最小化。利用*Help*下拉菜单也可以打开如图1.14所示的帮助窗口,选择不同的子菜单可显示不同的帮助信息。单击*MATLAB Help*子菜单打开的帮助窗口与单击问号图标所打开的帮助窗口相同。而单击*Using the Desktop*子菜单则将打开位于屏幕右侧的标题栏为*Desktop*的帮助窗口。



图1.14　*Help*窗口

单击*Using the Command Window*子菜单将打开屏幕右侧标题栏为Running Functions——Command Window and History的帮助窗口。如果要获得特定函数的帮助信息,则可单击帮助窗口左侧的*Index*标签页,并在下面显示的空白文本框内输入函数名,或在MATLAB命令窗口中输入以下信息:

```
help FunctionName
```

其中,*FunctionName*为要查询的函数名。通过帮助窗口的*Index*标签页可获得相似的帮助信息。通常,命令窗口无法获得方程式帮助信息。

1.5 使用 MATLAB 的一些建议

下面给出几点关于如何在 MATLAB 环境中高效创建程序和函数的建议。

- 尽可能使用帮助文件。这样将减少语法错误,以及由于不正确或不恰当地引用 MATLAB 函数而产生的错误。
- 在文本编辑器中编写程序和函数并保存为".m"文件。这将节省时间,减少代码。使用 MATLAB 编辑器/调试器时,将极大地简化调试过程。
- 尽量减少由程序或函数所组成表达式的数目。在易读性和简洁性之间采用折中的办法,并充分考虑函数和调用过程的执行速度和直接性。
- 程序和函数应尽可能使用图形化开发手段。这种方法可检测出隐含的代码错误,从而缩短代码的开发过程,并可使分析和模型化的物理过程变得容易理解。
- 最为重要的是,要采用另一种手段验证程序或函数输出结果的正确性。

1.6 本书的组织结构及目的

本书的目的在于帮助读者通过编写易读、简洁、正确的 MATLAB 程序,从而获得各个工程领域中物理模型和经验公式的数值解,并实现计算结果的图形化显示。

本书可用于以下几个方面:

- 学习 MATLAB 相关知识
- 作为工程类的教学用书
- 作为用数值方法求解多种工程问题的参考用书
- 作为获得大量 MATLAB 解决方案的资源库

本书适用于以下读者:熟悉微积分、线性代数及工程数学的相关知识;具有利用工程方法求解工程问题的能力;具有利用数学模型分析元件、设备和系统响应的能力。这是保证所编写程序正确执行的重要前提。本书分为两部分,第一部分包括第 1 章至第 7 章。介绍了 MATLAB 语法、命令及结构化编程的基本知识。第二部分包括第 8 章至第 14 章,详细介绍了如何利用第一部分知识获得工程问题数值解的方法。其中的部分章节介绍了如何利用 MATLAB 工具箱降低程序复杂性以获得不同难度工程问题的数值解。第 10 章详细介绍了控制工具箱和 Simulink 仿真器。第 13 章介绍了优化工具箱。第 14 章介绍了工程统计工具箱。第 8 章、第 11 章和第 12 章介绍了偏微分方程工具箱。第 5 章和第 9 章介绍了符号工具箱。

1.7 应用函数一览

表 1.2 给出了基本的算术函数。表 1.3 给出了三角函数和双曲函数。表 1.4 给出了特殊常量值。表 1.7 给出了 MATLAB 语法特殊符号。表 1.5 和表 1.6 分别给出了实现十进制数到整数转化的函数和复数操作函数。表 1.8 介绍了本章应用的其他常用函数。

表 1.8 第 1 章应用的 MATLAB 函数介绍

MATLAB 函数	说明
clc	清空命令窗口
clear	从工作空间(计算机内存)中删除变量
close all	关闭(删除)全部图形窗口
format	格式化显示命令窗口输出信息

练习

1.1 扇形的惯性力矩为①:

$$I=\left(\frac{\pi}{8}-\frac{8}{9\pi}\right)r^4$$

式中,r 为圆的半径。计算当 $r=2.5$ cm 时的 I 值。

1.2 压力盘簧曲率修正系数为②:

$$K=\frac{4c-1}{4c-4}+\frac{0.615}{c}$$

式中,$c=D/d$。D 为簧圈直径,d 为构成线圈的导线直径。求 $c=5$ 时的 K 值。

1.3 平板梯形板簧挠度形状因数为③:

$$K=\frac{3}{(1-B)^3}[0.5-2B+B(1.5-\ln B)]$$

式中,$B<1$ 为梯形末端比值,求 $B=0.6$ 时的 K 值。

1.4 两个半径分别为 R 和 r 的滑轮,中心距为 S,在滑轮间传动的皮带长度 L 可由下式给出④:

$$L=2S\cos\theta+\pi(R+r)+2\theta(R-r)$$

式中,

$$\theta=\sin^{-1}\left(\frac{R-r}{S}\right)$$

求 $R=30$ cm, $r=12$ cm, $S=50$ cm 时的 L 值。

1.5 闸块式制动器的扭矩 T 可由下式给出⑤:

$$T=\frac{4fF_nr\sin(\theta/2)}{\theta+\sin\theta}$$

式中,θ 为弧度表示的接触角,f 为摩擦系数,r 为鼓轮半径,F_n 为作用于鼓轮的作用力。求 $F=250$ N, $f=0.35$, $r=0.4$ m, $\theta=60°$时的 T 值。

1.6 截面边长分别为 A 和 B 的矩形风道中,空气流动阻力与直径为 D 的圆形风道中的

---

① 参见 T. G. Hicks, 出处同上, 第 8 页。
② 参见 T. G. Hicks, 出处同上,第 78 页。
③ 参见 T. G. Hicks, 出处同上,第 95 页。
④ 参见 T. G. Hicks, 出处同上,第 105~106 页。
⑤ 参见 T. G. Hicks, 出处同上,第 109 页。

空气流动阻力相同。圆形风道直径 D 可由下式给出[①]:

$$D = 1.265 \times \left[\frac{(AB)^3}{A+B}\right]^{1/5}$$

求 $A = 1.7\ \mathrm{m}, B = 1.2\ \mathrm{m}$ 时的 D 值。

1.7　有 n 个开槽的间歇工作轮的最大角加速度为[②]:

$$a_C = \omega^2 \frac{M(1-M^2)\sin\alpha}{(1+M^2-2M\cos\alpha)^2}$$

式中,

$$\cos\alpha = \sqrt{\left(\frac{1+M^2}{4M}\right)^2 + 2} - \left(\frac{1+M^2}{4M}\right)$$

$$M = \frac{1}{\sin(\pi/n)}$$

求 $n = 6$ 时的 a_C/ω^2 值。

1.8　标准条件下在钢管中流动的空气压降可由下式给出[③]:

$$\Delta p = \frac{0.03L}{d^{1.24}}\left(\frac{V}{1000}\right)^{1.84}$$

式中,L 为管长(单位:m),V 为空气流动速度(单位:m/min),d 为钢管直径(单位:mm)。求 $L = 3000\ \mathrm{m}, d = 45\ \mathrm{mm}, V = 1600\ \mathrm{m/min}$ 时的 Δp 值。

1.9　下列表达式[④] 表达了在主接触应力 F 的作用下,两个球体挤压到一起时,在 x,y 和 z 方向上所产生的接触应力值。

$$\sigma_x = \sigma_y = -p_{\max}\left[\left(1 - \frac{z}{a}\tan^{-1}\left(\frac{a}{z}\right)\right)(1-v_1) - 0.5\left(1+\frac{z^2}{a^2}\right)^{-1}\right]$$

$$\sigma_z = \frac{-p_{\max}}{1+z^2/a^2}$$

式中,

$$a = \sqrt[3]{\frac{3F}{8}\,\frac{(1-v_1^2)/E_1 + (1-v_2^2)/E_2}{1/d_1 + 1/d_2}}$$

$$p_{\max} = \frac{3F}{2\pi a^2}$$

v_j, E_j 和 $d_j (j = 1, 2)$分别为两个球体的波松比、弹性模量和直径。

求 $v_1 = v_2 = 0.3$, $E_1 = E_2 = 3 \times 10^7$ psi, $d_1 = 1.5$ in, $d_2 = 2.75$ in, $F = 100$ lb, $z = 0.01$ in 时的主应力值[⑤]。

[答案:$\sigma_x = -20\,002$ psi,$\sigma_z = -177\,120$ psi。]

---

① 参见 T. G. Hicks, 出处同上,第 165 页。
② 参见 T. G. Hicks, 出处同上,第 125 页。
③ 参见 T. G. Hicks, 出处同上,第 223 页。
④ 参见 J. E. Shigley and C. R. Mischke, *Mechanical Engineering Design*, 第五版, McGraw-Hill 纽约,1989 年。
⑤ 1 psi = 6.894 76 Pa;1 in = 2.54 cm;1 lb = 0.453 kg——编者注

1.10 下列表达式[①] 描述了在主接触应力 F 的作用下,两个平行圆柱体挤压到一起时在 x,y 和 z 方向所产生的接触应力值:

$$\sigma_x = -2v_2 p_{max}\left(\sqrt{1+\frac{z^2}{b^2}}-\frac{z}{b}\right)$$

$$\sigma_y = -p_{max}\left(\left(2-\left(1+\frac{z^2}{b^2}\right)^{-1}\right)\sqrt{1+\frac{z^2}{b^2}}-2\frac{z}{b}\right)$$

$$\sigma_z = \frac{-p_{max}}{\sqrt{1+z^2/b^2}}$$

$$\tau_{yz} = 0.5(\sigma_y - \sigma_z)$$

式中,

$$p_{max} = \frac{2F}{\pi bL}$$

$$b = \sqrt{\frac{2F}{\pi L}\frac{(1-v_1^2)/E_1-(1-v_2^2)/E_2}{1/d_1+1/d_2}}$$

v_j,E_j 和 $d_j(j=1,2)$分别为两个圆柱体的波松比、弹性模量和直径。

求 $v_1=v_2=0.3, E_1=E_2=3\times10^7$ psi, $d_1=1.5$ in, $d_2=2.75$ in, $F=100$ lb, $L=2$ in, $z=0.001$ in 时的主应力值。

[答案:$b=0.0014$ in, $p_{max}=23\ 251$ psi, $\sigma_x=-7085.7$ psi, $\sigma_y=-4843.8$ psi, $\sigma_z=-18\ 775$ psi。]

1.11 流体轴承的载荷数由下式给出[②]:

$$N_L = \frac{\pi\varepsilon\sqrt{\pi^2(1-\varepsilon^2)+16\varepsilon^2}}{(1-\varepsilon^2)^2}$$

式中,ε 为偏心率。

求 $\varepsilon=0.8$ 时的 N_L 值。

[答案:$N_L=72.022$。]

1.12 高为 h 的螺栓,材料的弹性模量为 E。通过直径为 d_0 的螺孔时,螺栓刚度 k 可由下式得出[③]:

$$k = \frac{\pi E d_0 \tan 30°}{\ln\left(\frac{(d_2-d_0)(d_1+d_0)}{(d_2+d_0)(d_1-d_0)}\right)}$$

其中,d_1 为孔下垫圈的直径,且:

$$d_2 = d_1 + h\tan 30°$$

求 $h=1.25$ in, $d_0=0.25$ in, $d_1=0.625$ in, $E=3\times10^7$ psi 时的 k 值。注意在 MAT-

① 参见 J. E. Shigley 和 C. R. Mischke, 出处同上。

② 参见 R. L. Norton, *Machine Design: An Integrated Approach*, Prentice Hall, Upper Saddle River, NJ, 1996 年。

③ 参见 A. H. Burr and J. B. Cheatham, *Mechanical Analysis and Design*, 第二版, Prentice Hall, Upper Saddle River. NJ, 1995 年,第 423 页。

LAB 中三角函数的参数单位应为弧度。

[答案：$d_2 = 1.3467$ in, $k = 2.8842 \times 10^7$ lb/in。]

1.13 一个长管,其内表面半径为 a,温度为 T_a;外表面半径为 b,温度为 T_b,则其径向和切向应力可分别表示为[①]:

$$\sigma_r = \frac{\alpha E(T_a - T_b)}{2(1-v)\ln(b/a)}\left[\frac{a^2}{b^2 - a^2}\left(\frac{b^2}{r^2} - 1\right)\ln\left(\frac{b}{a}\right) - \ln\left(\frac{b}{r}\right)\right]$$

$$\sigma_t = \frac{\alpha E(T_a - T_b)}{2(1-v)\ln(b/a)}\left[1 - \frac{a^2}{b^2 - a^2}\left(\frac{b^2}{r^2} + 1\right)\ln\left(\frac{b}{a}\right) - \ln\left(\frac{b}{r}\right)\right]$$

其中,r 为管子的径向坐标,E 为管子材料的弹性模量,α 为热膨胀系数。沿径向管壁的温度分布为:

$$T = T_b + \frac{(T_a - T_b)\ln(b/r)}{\ln(b/a)}$$

求 $\alpha = 1.2 \times 10^{-5}$ in/in/°F, $E = 3 \times 10^7$ psi, $v = 0.3$, $T_a = 500$ °F, $T_b = 300$ °F, $a = 0.25$ in, $b = 0.5$ in, $r = 0.375$ in 时的应力值及温度 T。

[答案：$\sigma_r = -8011.5$ psi, $\sigma_t = 5231.9$ psi, $T = 383.0075$。]

1.14 压力为 p_0 的气体,在可逆绝热条件下从容器中溢出,其质量与流量之比如下[②]:

$$\Psi = \sqrt{\frac{k}{k-1}}\sqrt{\left(\frac{p_e}{p_0}\right)^{2/k} - \left(\frac{p_e}{p_0}\right)^{(k+1)/k}}$$

式中,p_e 为容器出口的外部压力,k 为可逆绝热的气体常数。

求 $k = 1.4$, $p_e/p_0 = 0.3$ 时的 Ψ 值。

[答案：$\Psi = 0.4271$。]

1.15 抛物线型横截面开口管道流动的流量系数为[③]:

$$K = \frac{1.2}{x}\left[\sqrt{16x^2 + 1} + \frac{1}{4x}\ln\left(\sqrt{16x^2 + 1} + 4x\right)\right]^{-2/3}$$

式中,x 为最大水深与液体表面处管道宽度的比值。

求 $x = 0.45$ 时的 K 值。

[答案：$K = 1.3394$。]

1.16 依据以下给出的公式[④],当 $n = 0$ 时可在 10^{-7} 的误差范围内算出 π 的近似值,而当 n 取两项($n = 0, 1$)时其误差范围则为 10^{-15}。实际上,n 的取值每增加一个,π 近似值的精度将提高 10^{-8}。因此,n 取 4 项($n = 0, 1, 2, 3$)时 π 值可计算到 31 位,可用 MATLAB 符号工具箱进行验证。

$$\frac{1}{\pi} = \frac{\sqrt{8}}{9801}\sum_{n=0}^{\infty}\frac{(4n)!(1103 + 26\,390n)}{(n!)^4 396^{4n}}$$

注意,阶乘的计算可采用以下 gamma 函数:gamma$(n+1) = n!$, gamma$(4n+1) = (4n)!$, $n = 0, 1, \ldots$ 。

---

① 参见 A. H. Burr 和 J. B. Cheatham, 出处同上,第 496 页。

② 参见 W. Beitz 和 K. H. Kuttner, Eds., *Handbook of Mechanical Engineering*, Springer-Verlag, 纽约, 1994 年, 第 C15 页。

③ 参见 H. W. King, *Handbook of Hydraulics*, 第四版, McGraw-Hill, 纽约, 1954 年, 第 7~24 页。

④ 参见 S. Ramanujan, "Modular equations and approximations to π", *Quart. J. Math*, 45, 第 350~372 页。

1.17 冷空气标准狄塞尔循环的热效率可表示为①:

$$\eta = 1 - \frac{1}{r^{k-1}}\left[\frac{r_c^k - 1}{k(r_c - 1)}\right]$$

式中,r 为压缩比,r_c 为截止比,空气介质时 $k = 1.4$。

求 $r = 10, r_c = 3$ 时的 η 值。

1.18 缩张管中,任意段截面面积 A 与声速流段,即马赫数 $M = 1$ 段的截面面积 A^* 的比值如下②:

$$\frac{A}{A^*} = \frac{1}{M}\left[\left(\frac{2}{k+1}\right)\left(1 + \frac{k-1}{2}M^2\right)\right]^{(k+1)/[2(k-1)]}$$

式中,空气介质时,$k = 1.4$。

求 $M = 2$ 时的 A/A^* 值。

① 参见 M. J. Moran 和 H. N. Shapiro, *Fundamentals of Engineering Thermodynamics*, John Wiley and Sons, 纽约, 1992 年, 第 367 页。

② 参见 M. J. Moran 和 H. N. Shapiro, 出处同上, 第 418 页。

第2章　矩阵与 MATLAB

本章将通过向量和矩阵及其运算来说明 MATLAB 的基本语法。

2.1　概述

MATLAB 是一门语言,其运算指令和语法基于一系列基本的矩阵运算及其扩展运算。因此,为了更好地利用 MATLAB 语言的优越性和简洁性,首先对基本矩阵的定义、符号体系进行概述,然后给出几个应用实例。2.2 节、2.3 节和 2.4 节所论述的内容为基础内容,在本章及后续章节中将会大量应用。

2.2　矩阵和向量

由 m 行 n 列构成的数组 $\boldsymbol{A}$ 称为$(m\times n)$矩阵。总共由 $m\times n$ 个元素组成,并按以下形式排列:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{m1} & & & a_{mn} \end{bmatrix} \rightarrow (m\times n)$$

矩阵的元素记为 a_{ij},其中 i 表示行,j 表示列。

矩阵 $\boldsymbol{A}$ 的几种特例阐明如下。

方阵

当 $m=n$ 时,$\boldsymbol{A}$ 称为方阵。

对角阵

当 $a_{ij}=0, i\neq j$,且 $m=n$ 时,得到对角阵:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & & \\ \vdots & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix} \rightarrow (n\times n)$$

当 $a_{ii}=1$ 并且 $m=n$ 时,产生单位矩阵 $\boldsymbol{I}$,记做:

$$\boldsymbol{I}=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}$$

列矩阵和行矩阵(向量)

当 $a_{ij}=a_{i1}$(即只有 1 列)时,$\boldsymbol{a}=\boldsymbol{A}$ 称为列矩阵或者列向量,记做:

$$\boldsymbol{a}=\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}=\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}\rightarrow(m\times 1)$$

m 称为向量长度。

当 $a_{ij}=a_{1j}$(即矩阵只有 1 行)时,矩阵 $\boldsymbol{a}$ 称为行矩阵或者行向量,记做:

$$\boldsymbol{a}=[a_{11}\quad a_{12}\quad \cdots \quad a_{1n}]=[a_1\quad a_2\quad \cdots\quad a_n]\rightarrow(1\times n)$$

在 MATLAB 中,这是向量的默认定义。此时,n 称为向量长度。故向量可表示为一个行矩阵或一个列矩阵。

矩阵和向量的转置

矩阵的转置用(′)表示,定义如下:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{m1} & & & a_{mn} \end{bmatrix}\rightarrow(m\times n)$$

如果 $\boldsymbol{A}$ 是一个 $m\times n$ 的矩阵,则其转置矩阵 $\boldsymbol{W}=\boldsymbol{A}'$ 为以下$(n\times m)$的矩阵:

$$\boldsymbol{W}=\boldsymbol{A}'=\begin{bmatrix} w_{11}=a_{11} & w_{12}=a_{21} & \cdots & w_{1m}=a_{m1} \\ w_{21}=a_{12} & w_{22}=a_{22} & & \\ \vdots & & \ddots & \\ w_{n1}=a_{1n} & & & w_{nm}=a_{mn} \end{bmatrix}\rightarrow(n\times m)$$

即行数和列数进行了交换。

对于列向量和行向量,如果:

$$\boldsymbol{a}=\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}\rightarrow(m\times 1)\quad 则\quad \boldsymbol{a}'=[a_1\, a_2 \cdots a_m]\rightarrow(1\times m)$$

如果:

$$\boldsymbol{a}=[a_1\, a_2\cdots a_m]\rightarrow(1\times m)\quad 则\quad \boldsymbol{a}'=\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}\rightarrow(m\times 1)$$

2.3 向量的创建

向量可以表示为：

$$f=[a\ x\ b\ \cdots]\ 或\ f=[a,\ x,\ b,\ \cdots]$$

其中，a，x，b，…可以是变量、数值、表达式或字符串(参见 3.1 节)。如果是变量或表达式，则所有变量及由这些变量构成的表达式必须预先定义。并且在执行语句之前，每一个变量必须先赋值。表达式和变量可以任何组合和任何次序出现。对于采用下列形式的情况：

$$f=[a\ x\ b\ \cdots]$$

符号之间必须要用空格隔开。然而，对于采用下列形式的情况：

$$f=[a,\ x,\ b,\ \cdots]$$

则可以省略空格[①]。

应该着重说明一点，如果 a 是一个表达式，则表达式字符和运算符之间没有空格。例如，如果 $a=h+d^s$，则 f 可以写为：

$$f=[h+d\hat{}s\ x\ b\ \cdots]$$

或

$$f=[h+d\hat{}s,\ x,\ b,\ \cdots]$$

冒号

在 MATLAB 中给出了几种方法，用于对向量和矩阵中的元素进行赋值。矩阵的创建方法请参见 2.4 节。第一种方法是采用冒号来指定数值范围和相邻值的步长。第二种方法是指定数值的范围和期望的数值个数。在前一种方法中，步长最为重要。而在后一种方法中，数值的个数最为重要。利用冒号创建向量的方法如下：

$$\boldsymbol{x}=s:d:f\ 或\ \boldsymbol{x}=(s:d:f)\ 或\ \boldsymbol{x}=[s:d:f]$$

其中，

s——起始值或初始化值

d——增量或减量值

f——结束值或终值

因此，可产生以下的行向量 $\boldsymbol{x}$：

$$\boldsymbol{x}=[s,s+d,s+2d,\cdots,s+nd]$$

其中，$s+nd\leqslant f$。注意，创建向量 $\boldsymbol{x}$ 时没有直接指定 n 的数值。s，d 和 f 可以是数值、变量和表达式的任意组合。

向量 $\boldsymbol{x}$ 中元素的个数(即向量长度)由以下 MATLAB 函数给出：

```
length(x)
```

d 省略时，MATLAB 默认 $d=1$。即：

① MATLAB 有时具备几种不同的方法来描述同一数值，适当时将分别加以介绍。进行数值描述时，可依据其可读性选择不同的描述方法。

```
x=s:f
```

产生向量：

$$x=[s,s+1,s+2,\cdots,s+n]$$

其中，$s+n\leqslant f$。同样，s 和 f 可以是数值、变量和表达式的任意组合。

下面通过几个例证说明上述函数的应用。首先，创建取值范围从 0.2 到 1.0，步长为 0.1 的行向量。程序代码为：

```
x=0.2:0.1:1
n=length(x)
```

执行结果为：

```
x=
  0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000 1.0000
n =
  9
```

通过对行向量进行转置可得到列向量。则上述程序代码应修改为：

```
x=(0.2:0.1:1)'
n=length(x)
```

式中的圆括号是必需的。如果未采用圆括号（或方括号），则 MATLAB 将只对数值 1 进行转置操作。上面程序的执行结果为：

```
x=
  0.2000
  0.3000
  0.4000
  0.5000
  0.6000
  0.7000
  0.8000
  0.9000
  1.0000
n=
  9
```

如果希望创建取值范围为 0.2 至 1.0，步长为 0.12 的行向量，程序代码为：

```
x=0.2:0.12:1
n=length(x)
```

执行结果为：

```
x =
  0.2000 0.3200 0.4400 0.5600 0.6800 0.8000 0.9200
n=
  7
```

注意，此时最大值为 0.92，因为 $0.92+0.12=1.04>1$，已超出向量数值的变化范围。

如果要创建取值范围为 1 至 7，步长为 1 的行向量，则程序代码为：

```
x=1:7
n=length(x)
```

省略步长数值时,MATLAB 默认步长增量为 1。执行结果为:

```
x=
  1  2  3  4  5  6  7
n=
  7
```

因此,创建取值范围为 0.5 至 7,步长为 1 的行向量时,程序代码变为:

```
x=0.5:7
n=length(x)
```

执行结果为:

```
x=
  0.5000  1.5000  2.5000  3.5000  4.5000  5.5000  6.5000
n=
  7
```

创建等间隔值 *n*

另一方面,可以指定 n 为从 s 到 f 的等间隔值,如下所示:

```
x=linspace(s, f, n)
```

其中,增量或减量值 d 由 MATLAB 通过下式计算得出:

$$d=\frac{f-s}{n-1}$$

s 和 f 的值可以为正也可以为负,并且允许 $s>f$ 或 $s<f$。如果没有指定 n 的值,则 MATLAB 会给出一个默认值 $n=100$。因此,linspace 可创建以下向量:

$$\boldsymbol{x}=[s,s+d,s+2d,\cdots,f=s+(n-1)d]$$

如果希望创建从 -2 到 6.5 的 8 个等间隔值,则程序代码为:

```
x=linspace(-2, 6.5, 8)
```

执行结果为:

```
x=
  -2.0000  -0.7857  0.4286  1.6429  2.8571  4.0714  5.2857  6.5000
```

如果需要用对数等间隔,则:

```
x=logspace(s, f, n)
```

其中,10^s 为初值,10^f 为终值,d 的定义同上。因此,可创建以下向量:

$$\boldsymbol{x}=[10^s\ 10^{s+d}\ 10^{s+2d},\cdots,10^f]$$

没有给出 n 时,MATLAB 给出默认值 $n=50$。因此,如果创建从 1 至 100 的 5 个等对数间隔值,程序代码为:

```
x = logspace(0, 2, 5)
```

执行结果为:

```
x =
  1.0000  3.1623  10.0000  31.6228  100.0000
```

向量元素访问

下面说明如何访问向量中的元素,并对其进行算术运算。令:

$$b = [b_1\ b_2\ b_3 \cdots\ b_n]$$

即创建一个 1 行 n 列的向量 $\boldsymbol{b}$。在 MATLAB 中,通过写入下标的方式访问向量中的各元素,即 $b(j)$的数值为向量的第 j 个元素 b_j,其中 $j=1, 2, \cdots, n$ 代表向量 $\boldsymbol{b}$ 中的第 j 个元素。也就是说,写入 $b(3)$可获得向量 $\boldsymbol{b}$ 中的第 3 个元素 b_3 的值。由于 MATLAB 的编译器能够知道矩阵 $\boldsymbol{b}$ 为一个$(1\times n)$的矩阵,因此可以不用双下标表示。当 $\boldsymbol{b}$ 为前面定义的向量时,$b(3)$和 $b(1,3)$是一样的。但如果写成 $b(3,1)$,则系统会给出错误信息,因为该向量并没有定义第 3 行。

相反,如果令:

$$\boldsymbol{b} = [b_1\ b_2\ b_3 \cdots\ b_n]'$$

则会创建一个列向量,即一个$(n\times 1)$的矩阵。如果要确定向量中第 3 个元素的值,同样可写为 $b(3)$,MATLAB 会返回相应于 b_3 的值。这也等同于写成 $b(3,1)$。但如果写为 $b(1,3)$,系统会给出错误信息,因为该向量没有定义第 3 列。

向量元素运算

若想创建值为 $[-2, 1, 3, 5, 7, 9, 10]$ 的向量 $\boldsymbol{x}$,可通过下式实现:

```
x = [-2,1:2:9,10]
```

或

```
x = [-2, 1, 3, 5, 7, 9, 10]
```

这意味着向量中的元素分别为 $x_1=-2, x_2=1, x_3=3, x_4=5, x_5=7, x_6=9, x_7=10$ 且向量长度为 7。可通过 MATLAB 表达式 $x(j)$ 访问向量 $\boldsymbol{x}$ 中的元素,其中 $j=1, 2, \cdots, 7$。例如,表达式 $x(5)$的返回值为 7。可通过保留字 end 访问向量中的最后一个元素,如下所示:

```
x = [-2, 1:2:9, 10];
xlast = x(end)
```

执行后,命令窗口将显示以下结果:

```
xlast =
  10
```

向量与标量进行加减运算时,标量与向量中的每一个元素相加减。因此:

```
x = [-2, 1, 3, 5, 7, 9, 10];
z = x - 1
```

的结果为:

```
z=
  -3 0 2 4 6 8 9
```

但是对于乘法、除法和幂运算,其运算规则有严格的限制,详见2.6.2节。

另一方面,有时只是要修改向量中的某些元素。例如,令 $z=[-2\ 1\ 3\ 5\ 7\ 9\ 10]$。要求向量的第2个元素除以2,则可以写为:

```
z=[-2,1,3,5,7,9,10];
z(2)=z(2)/2;
z
```

运行结果为:

```
z=
  -2.0000  0.5000  3.0000  5.0000  7.0000  9.0000  10.0000
```

如果同时要求第3个和第4个元素乘以3减去1,则可以写为:

```
z=[-2,1,3,5,7,9,10];
z(2)=z(2)/2;
z(3:4)=z(3:4)*3-1;
z
```

程序运行结果为:

```
z=
  -2.0000  0.5000  8.0000  14.0000  7.0000  9.0000  10.0000
```

注意,向量中的其他元素保持不变。MATLAB 编译器将赋值语句 $z(3:4)=z(3:4)*3-1$ 编译为以下过程:向量元素 $z(3)$和 $z(4)$的当前值分别乘以3,之后以原始值乘以3所得的中间值减1,再将所得的最终结果值代替 $z(3)$和 $z(4)$的原始值。作为应用广泛的程序语句,这种语法非常适于编写简洁的程序代码。

此外,还有几种方法可以访问向量中的元素。8个元素的向量如下:

```
y=[-1,6,15,-7,31,2,-4,-5];
```

如果要创建一个由 y 的第3个到第5个元素组成的新向量 $\boldsymbol{x}$,可以写为:

```
y=[-1,6,15,-7,31,2,-4,-5];
x=y(3:5)
```

结果即创建了3个元素的向量:

```
x=
  15  -7  31
```

如果希望创建一个由 y 的前两个元素和后两个元素组成的向量 $\boldsymbol{x}$,可输入:

```
y=[-1,6,15,-7,31,2,-4,-5];
x=[y(1), y(2), y(7), y(8)]
```

此外,还可以先定义一个向量,然后按如下方法使用该向量:

```
y=[-1,6,15,-7,31,2,-4,-5];
index=[1,2,7,8];
x=y(index)
```

或采用更为简洁的表达方式：

```
y=[-1,6,15,-7,31,2,-4,-5];
x=y([1, 2, 7, 8])
```

后面的两种方法是非常有用的。对应于向量 *y* 有 8 个元素的向量 *z*：

```
y=[-1, 6, 15, -7, 31, 2, -4, -5];
z=[10, 20, 30, 40, 50, 60, 70, 80];
```

用 sort 函数将向量 *y* 按升序排列(即从最大负值到最大正值排列)，然后根据向量 *y* 中新的元素顺序重新排列向量 *z*。由 *Help* 文件可得到 sort 函数的调用格式为：

```
[ynew, indx]=sort(y)
```

其中，***ynew*** 是重新排列后的向量 ***y***，***indx*** 是一个由元素位置所构成的向量，表示排列后向量 *y* 中元素的原始位置号。因此，程序

```
y=[-1, 6, 15, -7, 31, 2, -4, -5];
z=[10, 20, 30, 40, 50, 60, 70, 80];
[ynew, indx]=sort(y)
znew=z(indx)
```

的执行结果为：

```
ynew=
  -7  -5  -4  -1  2  6  15  31
indx=
  4  8  7  1  6  2  3  5
znew=
  40  80  70  10  60  20  30  50
```

可以看出，*indx*(1)=4 表明 *ynew*(1)就是 *y*(4)。相对 *z* 而言，简单利用顺序向量 ***indx***，可定义 ***znew***。

可以通过 find 命令扩展这种功能。使用 find 命令可以找到满足用户指定条件或表达式的向量(或矩阵)元素的位置(不是元素值)。使用 find 命令在 *y* 中选出非正元素，组成一个新的向量 *s*，过程如下。在 MATLAB 中，关系运算符“ <= ”代表“≤”(参见表 4.1)。

```
y=[-1, 6, 15, -7, 31, 2, -4, -5];
indxx=find(y<=0)
s=y(indxx)
```

得出：

```
indxx=
  1  4  7  8
s=
  -1  -7  -4  -5
```

命令行可以写为以下紧凑的形式:

```
y=[-1, 6, 15, -7, 31, 2, -4, -5];
s=y(find(y<=0))
```

MATLAB 中隐含向量和矩阵说明的一个最大好处,是可以帮助用户实现对于一组值的一系列操作。例如,如果希望确定 sin(x)的十等分值,其中,x 在 $-\pi\leqslant x\leqslant\pi$ 的区间内变化,则 MATLAB 语句:

```
x=linspace(-pi, pi, 10);
y=sin(x)
```

将生成以下向量:

```
y=
  -0.0000  -0.6428  -0.9848  -0.8660  -0.3420  0.3420  0.8660  0.9848  0.6428
  0.0000
```

向量的最大值和最小值

MATLAB 提供了访问向量(或矩阵)极值的方法。函数 min 用于获得向量中的最小值及其在向量中所处的位置。函数 max 用于获得向量中的最大值及其在向量中所处的位置。因此,获得上述程序所创建向量的最大值和最小值,则有:

```
x=linspace(-pi, pi, 10);
y=sin(x);
[ymax, kmax]=max(y)
[ymin, kmin]=min(y)
```

执行结果为:

```
ymax=
  0.9848
kmax=
  8
ymin=
  -0.9848
kmin=
  3
```

即向量 y 的最大值为第 8 个元素 0.9848,最小值为第 3 个元素 -0.9848。

2.4 矩阵的创建

对于(4×3)的矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}\rightarrow(4\times3)$$

可以通过以下几种方法来创建。创建矩阵的基本语法为:

$$A = [a_{11}\ \ a_{12}\ \ a_{13};a_{21}\ \ a_{22}\ \ a_{23};a_{31}\ \ a_{32}\ \ a_{33};a_{41}\ \ a_{42}\ \ a_{43}]$$

其中,分号表示行的结束。每行必须具有相同的列数。这种表达式也可以采用更形象的描述方法:

$$\begin{aligned} A = [\,& a_{11}\, a_{12}\, a_{13}\,;\cdots \\ & a_{21}\, a_{22}\, a_{23}\,;\cdots \\ & a_{31}\, a_{32}\, a_{33}\,;\cdots \\ & a_{41}\, a_{42}\, a_{43}\,] \end{aligned}$$

其中,省略号(…)是必需的,用以表明表达式接续下一行。也可以通过在每一行的末尾处按下 *Enter* 键来代替采用省略号(…)的方式创建矩阵。此时,表达式为:

$$\begin{aligned} A = [\,& a_{11}\, a_{12}\, a_{13} \\ & a_{21}\, a_{22}\, a_{23} \\ & a_{31}\, a_{32}\, a_{33} \\ & a_{41}\, a_{42}\, a_{43}\,] \end{aligned}$$

创建矩阵的第四种方法是分别创建四个具有相同列数的行向量,然后将其组合为矩阵。此时有:

$$\begin{aligned} v_1 &= [a_{11}\, a_{12}\, a_{13}]; \\ v_2 &= [a_{21}\, a_{22}\, a_{23}]; \\ v_3 &= [a_{31}\, a_{32}\, a_{33}]; \\ v_4 &= [a_{41}\, a_{42}\, a_{43}]; \\ A &= [v_1\,;v_2\,;v_3\,;v_4] \end{aligned}$$

其中,前四行的分号用于省略命令窗口的回显。

以上几种形式中,a_{ij} 可以是数值、变量、表达式或字符串。如果是变量或表达式,则执行该语句之前一定要首先通过用户输入或前面执行的表达式,对变量或由变量组成的表达式进行赋值。表达式和变量可以任意组合形式出现。如果是字符串,则每一行中的字母个数应相同,参见 3.1 节。

在 MALAB 中创建以下矩阵:

$$A = \begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}$$

代码为:

```
A=[11, 12, 13, 14;21, 22, 23, 24;31, 32, 33, 34;41, 42, 43, 44]
```

执行结果为:

```
A=
    11  12  13  14
    21  22  23  24
    31  32  33  34
    41  42  43  44
```

代码也可以采用以下方式书写：

```
A=[11,12,13,14;…
   21,22,23,24;…
   31,32,33,34;…
   41,42,43,44]
```

或者，

```
A=[11,12,13,14
   21,22,23,24
   31,32,33,34
   41,42,43,44]
```

或者，

```
v1=[11,12,13,14];
v2=[21,22,23,24];
v3=[31,32,33,34];
v4=[41,42,43,44];
A=[v1;v2;v3;v4]
```

以上几种方法创建的矩阵与第一种方法相同。

矩阵的阶数由下式得出：

```
[m, n]=size(A)
```

其中，m 为行数，n 为列数。确定矩阵 A 为(4×4)阶矩阵的代码为：

```
A=[11, 12, 13, 14;21, 22, 23, 24;31, 32, 33, 34;41, 42, 43, 44];
[m, n]=size(A)
```

执行结果为：

```
m =
  4
n =
  4
```

应用 length 函数可以获得矩阵的列数。如果 A 为(2×4)阶矩阵，则：

```
A=[1 2 3 4; 5 6 7 8];
L=length(A)
```

执行结果为：

```
L=
  4
```

利用撇号(′)可实现矩阵的转置操作。矩阵 A 的转置为：

```
A=[11, 12, 13, 14;21, 22, 23, 24;31, 32, 33, 34;41, 42, 43, 44]'
```

执行结果为：

```
A =
  11  21  31  41
  12  22  32  42
  13  23  33  43
  14  24  34  44
```

特殊矩阵的创建

下面介绍 4 个函数用以创建具有特殊元素值的矩阵。

函数 ones(r, c)可实现创建所有元素值皆为 1 的($r \times c$)阶矩阵。用其可方便地代替表达式 one(1:r, 1:c) = 1 的功能。

函数 zeros(r, c)可实现创建($r \times c$)阶的零矩阵，其所有元素值皆为 0。用其可方便地代替表达式 zero(1:r, 1:c) = 0 的功能。

函数 diag(a)可实现创建($n \times n$)阶的对角阵，矩阵对角元素值由长度为 n 的向量 $\boldsymbol{a}$ 给出。该函数也可以实现提取($n \times n$)阶矩阵 $\boldsymbol{A}$ 的对角元素值的功能。即：

```
b = diag(A)
```

其中，$\boldsymbol{b}$ 为包含矩阵 $\boldsymbol{A}$ 且对角元素值的长度为 n 的向量。

函数 eye(n)可实现创建($n \times n$)阶的单位阵 $\boldsymbol{I}$。

下面说明以上四个函数的用法。创建元素值为 1 的(2×5)阶矩阵的程序代码为：

```
on = ones(2, 5)
```

执行结果为：

```
on =
  1  1  1  1  1
  1  1  1  1  1
```

创建元素值皆为 0 的(3×2)阶矩阵的程序代码为：

```
zer = zeros(3, 2)
```

执行结果为：

```
zer =
  0  0
  0  0
  0  0
```

创建(3×3)阶对角阵 $\boldsymbol{A}$，且矩阵元素 $a_{11} = 4, a_{22} = 9, a_{33} = 1$ 的程序代码为：

```
a = [4, 9, 1];
A = diag(a)
```

或采用更简洁的程序代码：

```
A = diag([4, 9, 1])
```

执行结果为:

```
A =
  4  0  0
  0  9  0
  0  0  1
```

另一方面,如果已经创建($n \times n$)阶矩阵,则函数 diag 可获取矩阵对角元素值。如果矩阵 A 为前面定义的(4×4)阶矩阵,则下式可获取其对角元素值:

```
A = diag([11, 12, 13, 14;21, 22, 23, 24;31, 32, 33, 34;41, 42, 43, 44])
```

执行结果可得:

```
A =
  11
  22
  33
  44
```

而且,通过下式可创建由矩阵 A 的对角元素值构成的对角阵:

```
A = [11, 12, 13, 14;21, 22, 23, 24;31, 32, 33, 34;41, 42, 43, 44];
Adiag = diag(diag(A))
```

执行结果为:

```
Adiag =
  11   0   0   0
   0  22   0   0
   0   0  33   0
   0   0   0  44
```

创建(3×3)阶单位阵的代码为:

```
A = eye(3)
```

执行结果为:

```
A =
  1  0  0
  0  1  0
  0  0  1
```

矩阵元素的运算

考虑下面(3×5)阶矩阵的结构:

$$A = \begin{bmatrix} 3 & 5 & 7 & 9 & 11 \\ 20.0 & 20.25 & 20.5 & 20.75 & 21.0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow (3 \times 5)$$

该矩阵可通过如下语句创建:

```
A = [3:2:11;linspace(20, 21, 5);ones(1, 5)]
```

创建后的矩阵如下：

```
A =
   3.0000    5.0000    7.0000    9.0000   11.0000
  20.0000   20.2500   20.5000   20.7500   21.0000
   1.0000    1.0000    1.0000    1.0000    1.0000
```

矩阵元素 $a_{ij}(i=1,2,3,j=1,2,\cdots,5)$的访问如图 2.1 所示。则：

```
A(1, 1)→3
```

表示第 1 行第 1 列元素 a_{11}。

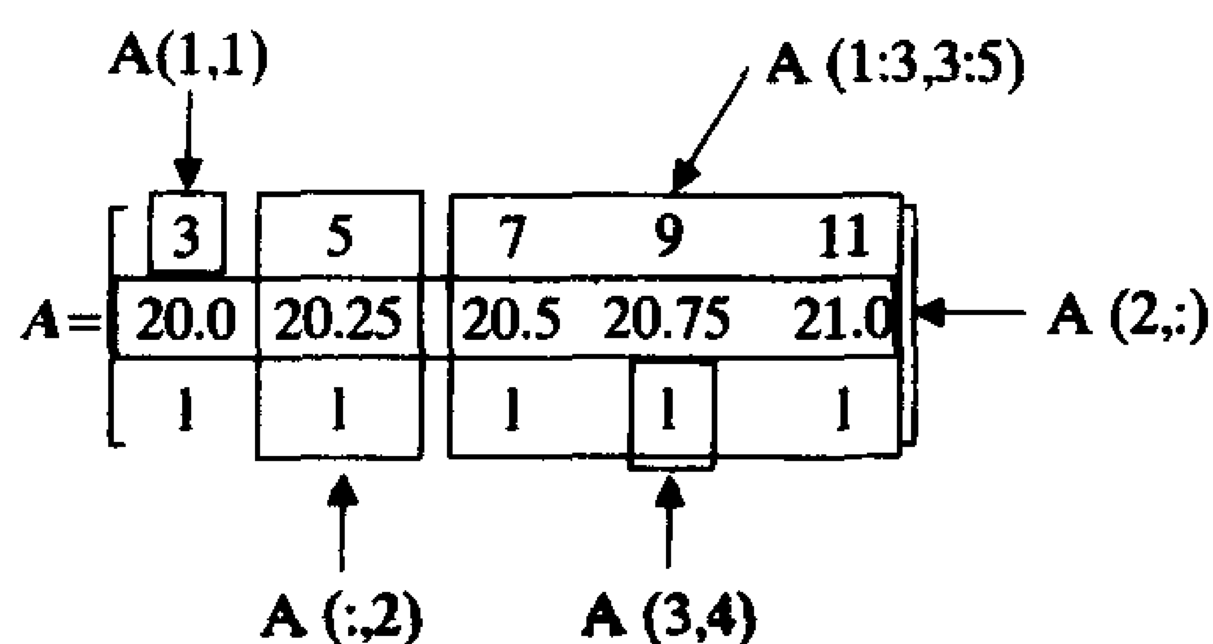


图 2.1　矩阵元素的访问

```
A(3, 4)→1
```

表示第 3 行第 4 列元素 a_{34}。

```
A(:, 2)→[5, 20.25, 1]'
```

可访问第 2 列中的所有元素 a_{12}，a_{22}和 a_{32}。其中，转置符号表明其为列向量。而

```
A(:, 2)
```

表示第 2 列中的所有元素。

```
A(2, :)→[20, 20.25, 20.5, 20.75, 21]
```

可访问第 2 行中的所有元素 a_{21}，a_{22}，a_{23}，a_{24}和 a_{25}。其中，

```
A(2, :)
```

表示第 2 行中的所有元素。

使用冒号可访问第 3 列到第 5 列，第 1 行到第 3 行元素构成的子矩阵。

```
A(1:3, 3:5)→[7, 9, 11;20.5, 20.75, 21;1, 1, 1]
```

表示一个(3×3)阶矩阵。书写矩阵 A 时，使用了默认步长 +1。因此，

```
A = [3:2:11;linspace(20, 21, 5);ones(1, 5)];
B = A(1:3, 3:5)
```

执行结果将生成一个(3×3)阶矩阵：

```
B =
   7.0000    9.0000   11.0000
  20.5000   20.7500   21.0000
   1.0000    1.0000    1.0000
```

创建一个大小与矩阵 *A* 相同但元素都等于 4 的矩阵,可由下面的表达式实现:

```
A=[3:2:11;linspace(20, 21, 5);ones(1, 5)];
[r, c]=size(A);
Z=4*ones(r, c)
```

结果生成:

```
Z=
  4  4  4  4  4
  4  4  4  4  4
  4  4  4  4  4
```

上述代码也可简化为:

```
A=[3:2:11;linspace(20, 21, 5);ones(1, 5)];
Z=4*ones(size(A))
```

可以采用类似于改变向量中元素的方法改变矩阵中的元素。函数 magic[①] 用于产生特殊矩阵,即矩阵中任意行或任意列中的元素之和,以及两个对角线上的元素之和都相等。对于 (4×4) 阶矩阵而言,其值为 34。令:

```
Z=magic(4)
```

结果显示:

```
Z=
  16   2   3  13
   5  11  10   8
   9   7   6  12
   4  14  15   1
```

实现第 2 行元素除以 2,第 2 列加到第 4 列并将结果值置于第 4 列的程序为:

```
Z=magic(4);
Z(2,:)=Z(2,:)/2;
Z(:,4)=Z(:,4) + Z(:,2);
Z
```

结果为:

```
Z=
  16.0000   2.0000   3.0000  15.0000
   2.5000   5.5000   5.0000   9.5000
   9.0000   7.0000   6.0000  19.0000
   4.0000  14.0000  15.0000  15.0000
```

使用下面的语句将矩阵 *Z* 中的对角元素置为 0:

```
Z=magic(4);
Z=Z-diag(diag(Z))
```

① 可获得 50 多个特殊矩阵。

结果为:

```
Z=
  0   2   3  13
  5   0  10   8
  9   7   0  12
  4  14  15   0
```

若将对角元素的值置为 5,可以采用下面的语句:

```
Z=magic(4);
Z=Z-diag(diag(Z))+5*eye(4)
```

结果为:

```
Z=
  5   2   3  13
  5   5  10   8
  9   7   5  12
  4  14  15   5
```

若将矩阵 Z 的对角元素分别置为 11,23,54 和 61,可以采用下面的语句:

```
Z=magic(4);
Z=Z-diag(diag(Z))+diag([11, 23, 54, 61])
```

结果为:

```
Z=
 11   2   3  13
  5  23  10   8
  9   7  54  12
  4  14  15  61
```

矩阵的最小值和最大值

函数 min 和 max 用于求取矩阵中各列的最小值和最大值。如对于 $(m \times n)$ 阶矩阵,函数 min 和 max 的输出向量长度为 n,向量元素值为矩阵中每一列的最小值或最大值。求取矩阵 magic(4)的最小值和最大值的程序代码为:

```
M=magic(4)
minM=min(M)
maxM=max(M)
```

执行结果得出:

```
M=
 16   2   3  13
  5  11  10   8
  9   7   6  12
  4  14  15   1
minM=
  4  2  3  1
```

```
maxM =
  16   14   15   13
```

如果求取全部元素的最大值,需要两次应用函数 max。即:

```
M = magic(4);
maxM = max(max(M))
```

结果为:

```
maxM =
  16
```

阵列运算

MATLAB 提供了两个创建矩阵的函数,通过复制标量、列向量、行向量和块矩阵来创建矩阵。这两个函数为:

```
repmat
```

和

```
meshgrid
```

函数 repmat 的调用格式为:

```
repmat(x, r, c)
```

其中,x 可以是标量、向量或矩阵,r 是 x 的行数,c 是 x 的列数。函数 repmat 非常适合于创建注释性输出,相关介绍参见第 3 章。2.6 节介绍了函数 meshgrid 在评估方面中的应用,第 7 章及后续章节介绍了该函数在三维表面中的显示应用。

下面介绍如何利用函数 repmat 从原始向量或矩阵创建不同的向量或矩阵。首先,介绍如何创建一个任意长度的列向量或行向量,且向量的所有元素具有相同的数值。要创建包含 6 个元素且每个元素值都为 45.72 的行向量 w,可输入:

```
w = repmat(45.72, 1, 6)
```

该表达式等同于:

```
w = [45.72, 45.72, 45.72, 45.72, 45.72, 45.72]
```

也可以用下列语句创建向量 w:

```
w(1, 1:6) = 45.72
```

如果要创建所有值全为 45.72 的(3×3)阶矩阵,可输入:

```
W = repmat(45.72, 3, 3)
```

或者:

```
W = [45.72, 45.72, 45.72; 45.72, 45.72, 45.72;45.72, 45.72, 45.72]
```

或者:

```
W(1:3, 1:3) = 45.72
```

每个表达式在 MATLAB 命令窗口都可生成矩阵：

```
W=
  45.7200  45.7200  45.7200
  45.7200  45.7200  45.7200
  45.7200  45.7200  45.7200
```

考虑向量：

$$s = [a_1\ a_2\ a_3\ a_4]$$

表达式

```
V = repmat(s, 3, 1)
```

可创建数值相等[①] 的矩阵：

$$V = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

创建的向量 s 有 3 行，每行有 4 列。表达式：

```
repmat(s, 3, 2)
```

则创建了以下数值相等的矩阵：

$$V = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

另一方面，命令：

```
V = repmat(s', 1, 3)
```

创建每列数值等于列向量 s'，共 3 列，每列 4 行的矩阵：

$$V = \begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \end{bmatrix}$$

表达式：

```
V = repmat(s', 2, 3)
```

创建以下数值相等的矩阵：

---

① 所谓数值相等，意味着在 MATLAB 中已为 v_{ij} 赋值。此处的表达形式是为了表明结果数组中元素的排列，并说明 repmat 函数的操作过程。

$$V=\begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \\ a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \end{bmatrix}$$

若有两个行向量 $\boldsymbol{s}$ 和 $\boldsymbol{t}$,则 MATLAB 表达式:

```
[U, V] = meshgrid(s, t)
```

与以下两条指令的结果相同:

```
U = repmat(s, length(t), 1)
V = repmat(t', 1, length(s))
```

每种情况下,$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 都是(length(t) × length(s))的矩阵。因此,如果:

$$\boldsymbol{s}=[s_1\ s_2\ s_3\ s_4]$$

$$\boldsymbol{t}=[t_1\ t_2\ t_3]$$

则命令:

```
[U, V] = meshgrid(s, t)
```

将生成两个(3×4)阶矩阵:

$$\boldsymbol{U}=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix} \tag{2.1a}$$

$$\boldsymbol{V}=\begin{bmatrix} t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 \\ t_3 & t_3 & t_3 & t_3 \end{bmatrix} \tag{2.1b}$$

函数 meshgrid 也可以用来仅返回一个矩阵,如:

```
W = meshgrid(s, t)
```

生成 $\boldsymbol{W}=\boldsymbol{U}$,其中,$\boldsymbol{U}$ 由式(2.1a)给出。例 2.5 给出了说明。

矩阵操作函数 fliplr(A)和 flipud(A)在某些应用中非常有用,可实现对矩阵行和列的旋转操作。考虑(2×5)阶矩阵:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix}\rightarrow(2\times5)$$

用以下语句创建:

$$\boldsymbol{A}=[a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15};a_{21}\ a_{22}\ a_{23}\ a_{24}\ a_{25}]$$

则,

$$\texttt{fliplr}(A) \rightarrow \begin{bmatrix} a_{15} & a_{14} & a_{13} & a_{12} & a_{11} \\ a_{25} & a_{24} & a_{23} & a_{22} & a_{21} \end{bmatrix} \rightarrow (2 \times 5)$$

$$\texttt{flipud}(A) \rightarrow \begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \end{bmatrix} \rightarrow (2 \times 5)$$

和

$$\texttt{flipud}(\texttt{fliplr}(A)) \rightarrow \begin{bmatrix} a_{25} & a_{24} & a_{23} & a_{22} & a_{21} \\ a_{15} & a_{14} & a_{13} & a_{12} & a_{11} \end{bmatrix} \rightarrow (2 \times 5)$$

函数 fliplr(A)与 flipud(A)所产生的结果同样可由矩阵下标中的冒号得到。例如：

```
C = fliplr(A)
```

产生与下式相同的结果：

```
C = A(:, length(A): - 1:1)
```

考虑向量：

$$\boldsymbol{C} = [A\ \texttt{fliplr}(A)]' \rightarrow \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{21} \end{bmatrix} \rightarrow (10 \times 2)$$

产生的第 5 行和第 6 行相同。如果要去掉其中的一行，可以通过表达式[ ]使其中的一行为空，方括号中没有空格。用表达式：

```
C(length(A),:) = []
```

或

```
C(length(A) + 1,:) = []
```

可以使矩阵 $\boldsymbol{C}$ 缩减为：

$$\boldsymbol{C} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{21} \end{bmatrix} \rightarrow (9 \times 2)$$

其中,C 为(9×2)的矩阵。表达式 `C(length(A), :) = []`表示矩阵 C 中第 `length(A)`行中所有元素的值被指定为[](这种情况下,相当于值被删除)。尽管知道 A 的长度为5,但还是使用函数 `length(A)`让 MATLAB 计算为好。

通过 MATLAB 三种不同操作的结果来进一步说明上述用法。首先,创建两个(2×5)阶矩阵 $\boldsymbol{A}$ 和$\boldsymbol{B}$:

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix}$$

下面,考虑在三种 MATLAB 操作中的用法:

加/减: $C = A \pm B$

$$\boldsymbol{C}=\begin{bmatrix} a_{11}\pm b_{11} & a_{12}\pm b_{12} & a_{13}\pm b_{13} & a_{14}\pm b_{14} & a_{15}\pm b_{15} \\ a_{21}\pm b_{21} & a_{22}\pm b_{22} & a_{23}\pm b_{23} & a_{24}\pm b_{24} & a_{25}\pm b_{25} \end{bmatrix}\rightarrow(2\times5)$$

C 为(2×5)阶矩阵。

列扩展: $C = [A,\ B]$

$$\boldsymbol{C}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix}\rightarrow(2\times10)$$

C 为(2×10)阶矩阵。

行扩展: $C = [A;B]$

$$\boldsymbol{C}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix}\rightarrow(4\times5)$$

C 为(4×5)阶矩阵。

更进一步,如果:

$$\boldsymbol{x}=[x_1\ x_2\ x_3]$$

$$\boldsymbol{y}=[y_1\ y_2\ y_3]$$

则,

```
Z=[x',y']
```

或

```
Z=[x;y]'
```

可生成:

$$\boldsymbol{Z}=\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}\rightarrow(2\times3)$$

然而,

```
Z=[x';y']
```

可生成：

$$Z=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \rightarrow (6\times 1)$$

在按某种顺序存放数据时，这两种变换是非常有用的。

下面说明向量 $\boldsymbol{a}=[1,\ 2,\ 3]$和 $\boldsymbol{b}=[4,\ 5,\ 6]$的计算结果。代码：

```
x=[1,2,3];
y=[4,5,6];
Z1=[x,'y']
Z2=[x;y]'
Z3=[x';y']
```

执行结果为：

```
Z1 =
   1   4
   2   5
   3   6
Z2 =
   1   4
   2   5
   3   6
Z3 =
   1
   2
   3
   4
   5
   6
```

2.5　点运算

本节介绍 MATLAB 中的点运算(.)。在 MATLAB 中点运算是对同阶矩阵中逐个元素进行的算术运算。考虑下面的(3×4)阶矩阵：

$$X=\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}$$

和

$$M=\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$$

直接写出 MATLAB 点运算：

$$Z_m = X.*M = \begin{bmatrix} x_{11}*m_{11} & x_{12}*m_{12} & x_{13}*m_{13} & x_{14}*m_{14} \\ x_{21}*m_{21} & x_{22}*m_{22} & x_{23}*m_{23} & x_{24}*m_{24} \\ x_{31}*m_{31} & x_{32}*m_{32} & x_{33}*m_{33} & x_{34}*m_{34} \end{bmatrix} \tag{2.2a}$$

$$Z_d = X./M = \begin{bmatrix} x_{11}/m_{11} & x_{12}/m_{12} & x_{13}/m_{13} & x_{14}/m_{14} \\ x_{21}/m_{21} & x_{22}/m_{22} & x_{23}/m_{23} & x_{24}/m_{24} \\ x_{31}/m_{31} & x_{32}/m_{32} & x_{33}/m_{33} & x_{34}/m_{34} \end{bmatrix} \tag{2.2b}$$

$$Z_e = X.\hat{}M = \begin{bmatrix} x_{11}\hat{}m_{11} & x_{12}\hat{}m_{12} & x_{13}\hat{}m_{13} & x_{14}\hat{}m_{14} \\ x_{21}\hat{}m_{21} & x_{22}\hat{}m_{22} & x_{23}\hat{}m_{23} & x_{24}\hat{}m_{24} \\ x_{31}\hat{}m_{31} & x_{32}\hat{}m_{32} & x_{33}\hat{}m_{33} & x_{34}\hat{}m_{34} \end{bmatrix} \tag{2.2c}$$

注意，点运算符号必须放在乘号、除号和指数运算符号的前面，即 * →. * ;/→./;和^→.^。对于矩阵的加法和减法来说，点运算是没必要的，参见式(2.6)。

对于点乘来说，如果 $X=x_0$，其中 x_0 为标量常数，则点乘和乘法运算是一样的，可写为：

$$Z_m = x_0 * M$$

类似地，当 $M=m_0$，即标量常数时，则可以写为：

$$Z_m = X * m_0$$

在这两种情况下，点操作都是不需要的。

对于点除运算，当 $M=m_0$ 且 m_0 为标量常数时，则有：

$$Z_d = X/m_0$$

此时点运算是不必要的。然而，当 $X=x_0$ 且 x_0 为标量常数时，则有：

$$Z_d = x_0./M$$

这里的点运算是必需的。

对于指数运算而言，无论 $M=m_0$ 为标量常数还是 $X=x_0$ 为标量常数时，点运算都是必需的。即：

$$Z_e = x_0.\hat{}\ M$$

和

$$Z_e = X.\hat{}\ m_0$$

例 2.1　指数向量的生成

为说明指数的点运算，考虑计算 2^j，其中 $j=1, 2, \cdots, 8$。程序为：

```
x = 1:8;
y = 2.^x
```

结果为：

```
y=
  2  4  8  16  32  64  128  256
```

因此,在 MATLAB 中,放在指数操作符(^)前面的小数点意味着以 2 为底,在 x 的每个取值处计算其指数值,并将计算结果作为向量 y 的元素值。上面的程序可简写为:

```
y=2.^(1:8)
```

其中,小括号是必需的。或写为:

```
y=2.^[1:8]
```

其中,方括号是必需的。

反之,如果求取 j^2 的值,则程序为:

```
y=(1:8).^2  %或 y=[1:8].^2
```

结果为:

```
y=
  1  4  9  16  25  36  49  64
```

如果令 $f(Y)$代替任意一个函数,如 $f(Y)=\sin(Y)$或 $f(Y)=\cosh(Y)$等,则当矩阵 $\boldsymbol{Y}$ 为一个(3×4)阶矩阵时:

$$\boldsymbol{Z}=f(Y)=\begin{bmatrix} f(y_{11}) & f(y_{12}) & f(y_{13}) & f(y_{14}) \\ f(y_{21}) & f(y_{22}) & f(y_{23}) & f(y_{24}) \\ f(y_{31}) & f(y_{32}) & f(y_{33}) & f(y_{34}) \end{bmatrix}$$

在函数阶数相同的情况下,可实现与点运算的联合运算。例如,如果 a,b,c,d 和 g 都是(3×2)阶矩阵且已被赋值,则表达式:

$$\boldsymbol{Z}=\left[\tan a-g\left(\frac{b}{c}\right)^d\right]^2$$

可写为:

```
Z=(tan(a)-g.*(b./c).^d).^2;
```

$\boldsymbol{Z}$ 中各元素的数值可通过以下的表达式计算得出:

$$\boldsymbol{Z}=\begin{bmatrix} (\tan(a_{11})-g_{11}*(b_{11}/c_{11})\hat{\ }d_{11})\hat{\ }2 & (\tan(a_{12})-g_{12}*(b_{12}/c_{12})\hat{\ }d_{12})\hat{\ }2 \\ (\tan(a_{21})-g_{21}*(b_{21}/c_{21})\hat{\ }d_{21})\hat{\ }2 & (\tan(a_{22})-g_{22}*(b_{22}/c_{22})\hat{\ }d_{22})\hat{\ }2 \\ (\tan(a_{31})-g_{31}*(b_{31}/c_{31})\hat{\ }d_{31})\hat{\ }2 & (\tan(a_{32})-g_{32}*(b_{32}/c_{32})\hat{\ }d_{32})\hat{\ }2 \end{bmatrix}$$

例 2.2　表达式的点运算

下面说明表达式的点运算。求在 $0\leqslant t\leqslant 1$ 区间内,t 取 6 等分值时表达式

$$v=\mathrm{e}^{-a_1 t}\frac{\sin(b_1 t+c_1)}{t+c_1}$$

的值。设 $a_1=0.2, b_1=0.9, c_1=\pi/6$,程序代码为:

```
a1=0.2;b1=0.9;c1=pi/6;
t=linspace(0, 1, 6);
v=exp(-a1*t).*sin(b1*t+c1)./(t+c1)
```

执行结果为:

```
v=
  0.9549  0.8590  0.7726  0.6900  0.6097  0.5316
```

注意,a_1,b_1 和 c_1 都是标量值且表达式只包含乘除操作,因此可不必使用点运算。

点运算的另一个应用是函数 meshgrid。语句中利用了向量 $s=[s_1\ s_2\ s_3\ s_4]$和 $t=[t_1\ t_2\ t_3]$,下列语句:

```
[U, V]=meshgrid(s, t)
```

可生成两个(3×4)阶矩阵:

$$\boldsymbol{U}=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix} \quad 和 \quad \boldsymbol{V}=\begin{bmatrix} t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 \\ t_3 & t_3 & t_3 & t_3 \end{bmatrix} \tag{2.3}$$

若让 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 中的相关元素相乘,则点乘:

$$\boldsymbol{Z}=\boldsymbol{U}.*\boldsymbol{V}$$

可生成以下结果[参见式(2.2a)]:

$$\boldsymbol{Z}=\begin{bmatrix} s_1*t_1 & s_2*t_1 & s_3*t_1 & s_4*t_1 \\ s_1*t_2 & s_2*t_2 & s_3*t_2 & s_4*t_2 \\ s_1*t_3 & s_2*t_3 & s_3*t_3 & s_4*t_3 \end{bmatrix} \tag{2.4}$$

$\boldsymbol{Z}$ 中的元素是向量 $\boldsymbol{s}$ 和 $\boldsymbol{t}$ 中元素的所有可能乘积值。进行加、减、除和指数运算时,结果相似。只不过是将乘法符号(*)用其他运算符替代而已。

例 2.3 极坐标到笛卡儿坐标的变换

下面说明如何利用函数 meshgrid 通过关系式:

$$x=r\cos\theta$$
$$y=r\sin\theta$$

实现图 2.2 所示的极坐标到笛卡儿坐标的变换。令 r 在 $0.5\leqslant r\leqslant 1$ 区间内取 3 个值,θ 在 $0\leqslant\theta\leqslant\pi/2$ 区间内取 4 个值。程序代码为:

```
rr=linspace(0.5,1,3);
thet=linspace(0,pi/2,4);
[r,theta]=meshgrid(rr,thet);
x=r.*cos(theta)
y=r.*sin(theta)
```

执行结果为:

```
x=
  0.5000  0.7500  1.0000
  0.4330  0.6495  0.8660
  0.2500  0.3750  0.5000
  0.0000  0.0000  0.0000
```

```
y=
      0        0        0
 0.2500   0.3750   0.5000
 0.4330   0.6495   0.8660
 0.5000   0.7500   1.0000
```

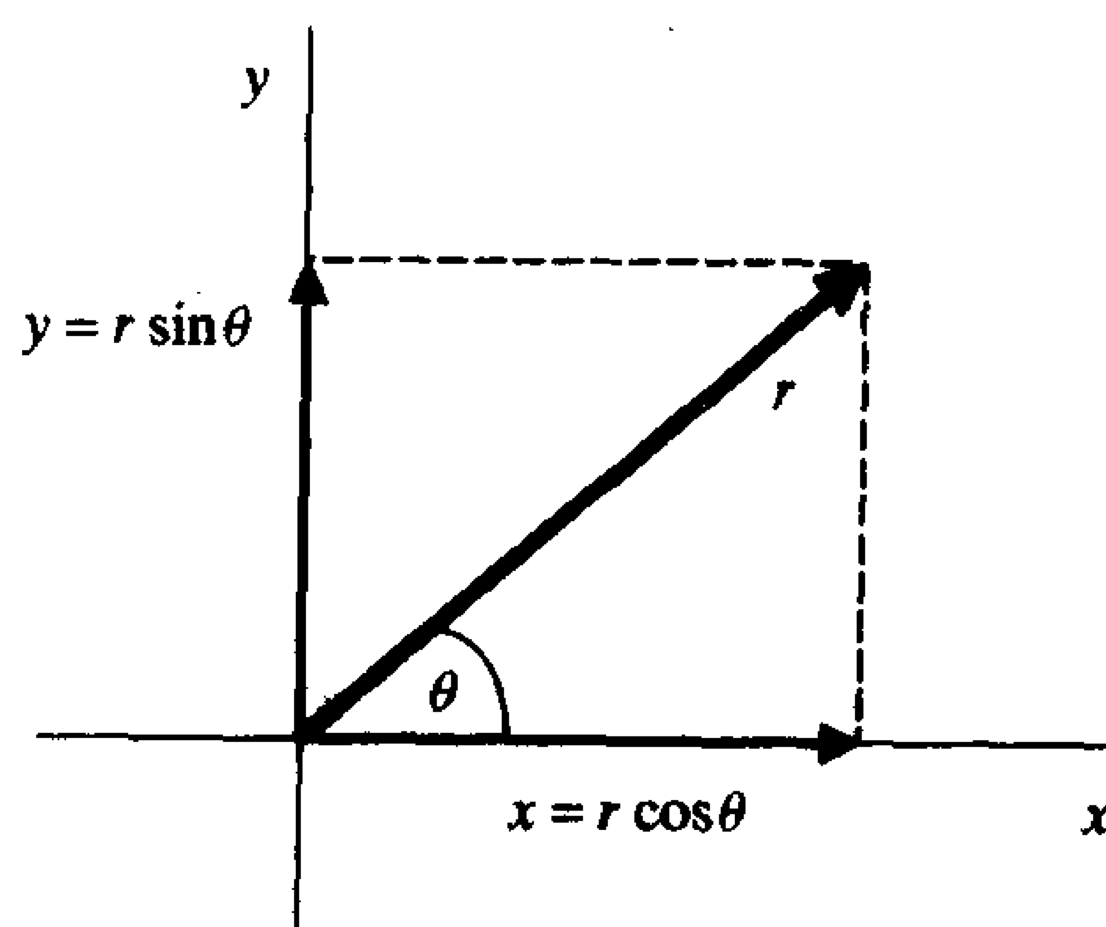


图 2.2　极坐标到笛卡儿坐标的变换

注意，$\theta=0, y=0$ 构成阵列 y 的第一行，对应的 x 值为 $x=0.5, 0.75$ 和 1，为阵列 x 的第一行显示值。例 2.8 给出了获得上述结果的其他方法。

下面说明求和函数

```
sum
```

和累积求和函数

```
cumsum
```

都经常用于点操作之中。

首先，介绍 sum 函数。当 $v=[v_1, v_2, \cdots, v_n]$ 时，则：

$$S = \text{sum}(v) \to \sum_{k=1}^{\text{length}(v)} v_k$$

其中，S 为标量值。当参数为矩阵时，函数将矩阵中的元素按列相加，并返回一个长度等于原始矩阵列数的行向量。因此，如果 $\boldsymbol{Z}$ 是一个 (3×4) 阶矩阵且其元素为 z_{ij}，则：

$$S = \text{sum}(Z) \to \left[\sum_{n=1}^{3} z_{n1} \quad \sum_{n=1}^{3} z_{n2} \quad \sum_{n=1}^{3} z_{n3} \quad \sum_{n=1}^{3} z_{n4}\right] \to (1\times 4) \tag{2.5}$$

为一个 3 元素向量。计算阵列中所有元素值时，则有：

$$S = \text{sum}(\text{sum}(Z)) \to \sum_{n=1}^{3} z_{n1} + \sum_{n=1}^{3} z_{n2} + \sum_{n=1}^{3} z_{n3} + \sum_{n=1}^{3} z_{n4} \to \sum_{i=1}^{4}\sum_{n=1}^{3} z_{ni} \to (1\times 1)$$

例 2.4　数组求和

为进一步说明如何应用函数 sum，求下列表达式的值：

$$z = \sum_{m=1}^{4} m^m$$

程序代码为:

```
m = 1:4;
z = sum(m.^m)
```

执行结果为:

```
z =
  288
```

代码可简写为:

```
z = sum((1:4).^(1:4))
```

例2.5 正态累积分布函数

当 $0\leqslant X\leqslant x$ 时,正态累积概率分布函数的近似值为[①]:

$$P(x) = P(X\leqslant x)\cong 1-\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\sum_{m=1}^{5}b_m(1+0.231\,641\,9x)^{-m}\quad 0\leqslant x\leqslant\infty$$

其中,$0.5\leqslant P(x)\leqslant 1$,且:

$b_1 = 0.319\,381\,530$
$b_2 = -0.356\,563\,782$
$b_3 = 1.781\,477\,937$
$b_4 = -1.821\,255\,978$
$b_5 = 1.330\,274\,429$

$1-P(|x|)$确定区间 $-\infty\leqslant x\leqslant 0$,其中 $0\leqslant 1-P(|x|)\leqslant 0.5$。

现在,计算并绘制区间 $-3\leqslant x\leqslant 3$ 内增量 $\Delta x=0.2$ 时的累积分布图。由题意可知,m 为(1×5)阶矩阵,x 为 $0\leqslant x\leqslant 3$ 区间内的(1×16)阶矩阵。向量和矩阵的长度可在第1章介绍的工作空间窗口中获得。首先,利用 meshgrid 函数由向量创建(16×5)阶矩阵以进行点运算及求和运算。但应用 sum 函数时,应将形成的矩阵进行转置,因为 sum 函数是按列进行矩阵求和运算的。本题中,希望对5项进行求和运算。首先,在区间 $0\leqslant x\leqslant 3$ 内进行相关运算,即可实现在 $-3\leqslant x\leqslant 0$ 区间内的运算。

程序代码为:

```
b = [0.319381530, -0.356563782, 1.781477937, ...
    -1.821255978, 1.330274429];                                  %(1×5)
m = 1:length(b);                                                 %(1×5)
x = 0:0.2:3;                                                     %(1×16)
[mm, Xm] = meshgrid(m, (1./(1 + 0.231641 * x)));                 %(16×5)
bmx = meshgrid(b, x);                                            %(16×5)
Px = 1 - exp(-0.5 * x.^2).* sum((bmx.*(Xm.^mm))')/sqrt(pi * 2);  %(1×16)
plot(x, Px, -fliplr(x), fliplr(1 - Px))
```

6.2节介绍的 plot 函数需要两对坐标参数,第一对参数给出从0到3的 $P(x)$值。第二对参数绘制从0到 -3 的 $1-P(x)$图形,但这些负数值并没有真正计算出 $P(x)$的值,因此需要进一步利用 fliplr 函数进行处理。表达式 -fliplr(x)等同于创建一个

① 参见 M. Abramowitz 和 I.A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Applied Mathematics Series 55, U.S. Government Printing Office, Washington D.C., 1964年,第932页。

新的向量 $x = -3:0.2:0$,而表达式 fliplr(1 - Px)将向量 $1 - P(x)$中的元素次序进行了颠倒,并创建了一个向量,其所有元素值都与 - fliplr(x)表达式给出的 x 的负数值相对应。程序执行结果如图 2.3 所示。

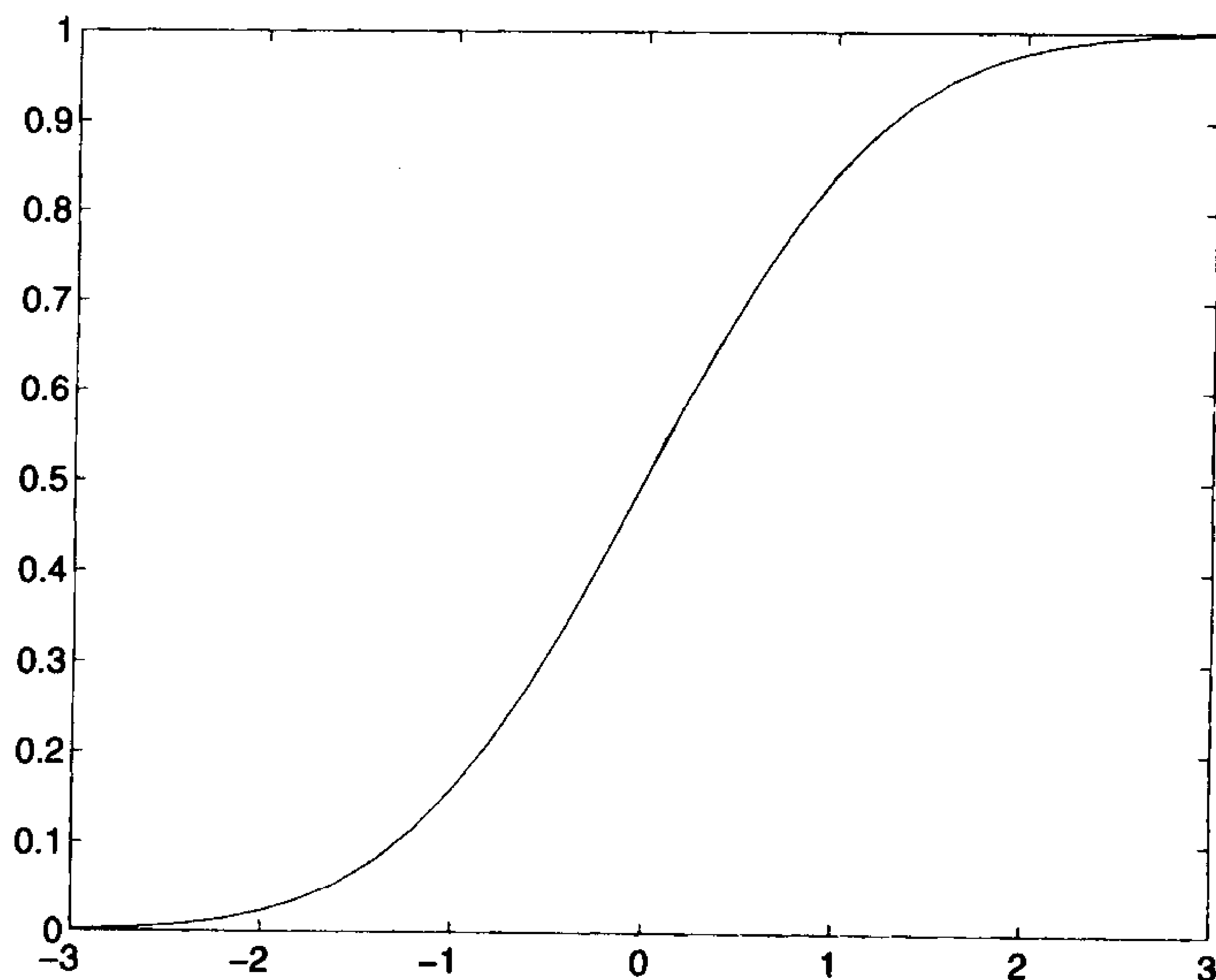


图 2.3　正态累积概率分布

向量 $\boldsymbol{v}$ 有 n 个元素 v_j,cumsum 函数对其操作生成另一个长度为 n 的向量,其元素为:

$$y = \text{cumsum}(v) \to \left[\sum_{k=1}^{1} v_k \quad \sum_{k=1}^{2} v_k \quad \cdots \quad \sum_{k=1}^{n} v_k\right] \to (1 \times n)$$

另一方面,如果 $\boldsymbol{W}$ 是由元素 w_{ij}组成的一个$(m \times n)$阶矩阵,则 cumsum($\boldsymbol{W}$)为以下矩阵:

$$\boldsymbol{Y} = \text{cumsum}(\boldsymbol{W}) = \begin{bmatrix} \sum_{k=1}^{1} w_{k1} & \sum_{k=1}^{1} w_{k2} & \cdots & \sum_{k=1}^{1} w_{kn} \\ \sum_{k=1}^{2} w_{k1} & \sum_{k=1}^{2} w_{k2} & & \\ \vdots & & \ddots & \\ \sum_{k=1}^{m} w_{k1} & & & \sum_{k=1}^{m} w_{kn} \end{bmatrix} \to (m \times n)$$

例 2.6　级数的收敛

考虑级数:

$$S = \sum_{n=1}^{10} \frac{1}{n^2}$$

求其前 10 项的收敛速度。程序为:

```
S = cumsum(1./(1:10).^2)
```

执行结果为:

```
S=
  1.0000  1.2500  1.3611  1.4236  1.4636  1.4914  1.5118  1.5274  1.5398
  1.5498
```

S 的每个元素值都是其前 n 项和,即 $S(1)$为第 1 项的值,$S(2)$为前 2 项的和值,依次类推。

例 2.7 双曲正割计算

计算下式[①] 求取双曲正割值。式中,$N=305$,且 x 在 $0\leqslant x\leqslant 2$ 的区间内按 5 等分取值。将计算值与其理论值做一比较。

$$\operatorname{sech} x = 4\pi \sum_{n=1,3,5}^{N\to\infty} \frac{n(-1)^{(n-1)/2}}{(n\pi)^2+4x^2}$$

进行点操作前,首先应利用 meshgrid 函数对求和指针向量 $\boldsymbol{n}$ 及参数向量 $\boldsymbol{x}$ 进行转换操作,生成等阶次矩阵。

程序为:

```
nn=1:2:305;                                                  %(1×153)
xx=linspace(0,2,5);                                          %(1×5)
[x,n]=meshgrid(xx,nn);                                       %(153×5)
s=4*pi*sum(n.*(-1).^((n-1)/2)./((pi*n).^2+4*x.^2));         %(1×5)
se=sech(xx);                                                 %(1×5)
compare=[s' se']                                             %(5×2)
```

程序运行后,在 MATLAB 命令窗口中显示结果:

```
compare=
  1.0021  1.0000
  0.8889  0.8868
  0.6501  0.6481
  0.4272  0.4251
  0.2679  0.2658
```

其中,右侧数值为理论值。由于 sum 函数可实现行元素的按列求和,所以选定 meshgrid 函数的参数顺序,生成(153×5)阶矩阵。

2.6 矩阵的数学运算

本节介绍矩阵的几种运算:加、减、乘、转置、行列式、方程式及其根(特征值)。这些运算经常用来求解各种工程问题。

2.6.1 加法和减法

如果有两个$(m\times n)$阶矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$,则有:

$$\boldsymbol{A}\pm\boldsymbol{B}=\begin{bmatrix} a_{11}\pm b_{11} & a_{12}\pm b_{12} & \cdots & a_{1n}\pm b_{1n} \\ a_{21}\pm b_{21} & a_{22}\pm b_{22} & & \\ \vdots & & \ddots & \\ a_{m1}\pm b_{m1} & & & a_{mn}\pm b_{mn} \end{bmatrix} \to (m\times n) \tag{2.6}$$

① 参见 L. B. W. Jolley, *Summation of Series*, 第二版, Dover Publications, New York, 1961 年。

2.6.2 乘法

如果有$(m \times k)$阶矩阵 $\boldsymbol{A}$ 和$(k \times n)$阶矩阵 $\boldsymbol{B}$,则:

$$\boldsymbol{C} = \boldsymbol{AB} = \begin{bmatrix} \sum_{j=1}^{k} a_{1j}b_{j1} & \sum_{j=1}^{k} a_{1j}b_{j2} & \cdots & \sum_{j=1}^{k} a_{1j}b_{jn} \\ \sum_{j=1}^{k} a_{2j}b_{j1} & \sum_{j=1}^{k} a_{2j}b_{j2} & & \vdots \\ \vdots & & \ddots & \\ \sum_{j=1}^{k} a_{mj}b_{j1} & \cdots & & \sum_{j=1}^{k} a_{mj}b_{jn} \end{bmatrix} \rightarrow (m \times n) \tag{2.7}$$

其中,$\boldsymbol{C}$ 为$(m \times n)$阶矩阵。应注意到,只有当两个矩阵相邻阶数(本例中为 k)相等时,矩阵乘积的定义才成立。换句话说,只能是$(m \times k)(k \times n) \rightarrow (m \times n)$。式(2.7)表明是对 k 项进行了求和操作。MATLAB 中矩阵乘积的表达式为:

```
C = A * B
```

当 $m = n$ 时,通常情况下 $\boldsymbol{AB} \neq \boldsymbol{BA}$。如果 $\boldsymbol{C} = \boldsymbol{AB}$,则 $\boldsymbol{C}$ 的转置为:

$$\boldsymbol{C}' = (\boldsymbol{AB})' = \boldsymbol{B}'\boldsymbol{A}'$$

同样,如果 $\boldsymbol{A}$ 为单位矩阵$(\boldsymbol{A} = \boldsymbol{I})$且 $m = n$,则:

$$\boldsymbol{C} = \boldsymbol{IB} = \boldsymbol{BI} = \boldsymbol{B}$$

例如,将下面两个矩阵相乘,并给出按上面两种方法得出的转置矩阵:

$$\boldsymbol{A} = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 11 & 12 \\ 21 & 22 \\ 31 & 32 \end{bmatrix}$$

程序为:

```
A = [11, 12, 13;21, 22, 23];
B = [11, 12;21, 22;31, 32];
C = A * B
Ctran1 = C'
Ctran2 = B' * A'
```

执行结果为:

```
C =
    776     812
    1406    1472
Ctran1 =
    776     1406
    812     1472
Ctran2 =
    776     1406
    812     1472
```

为求证矩阵相乘的结果,考虑下列级数[①]:

$$w(x,y) = \sum_{j=1}^{k} f_j(x) g_j(y)$$

x,y 具有一系列的值,假设 $x = x_1, x_2, \cdots, x_m$,而 $y = y_1, y_2, \cdots, y_n$。则 $w(x, y)$的值表示为:

$$w(x_i, y_j) = \sum_{l=1}^{k} f_l(x_i) g_l(y_j) \quad i = 1,2,\cdots,m \quad j = 1,2,\cdots,n$$

可将其作为$(m \times n)$阶矩阵 W 的一个元素。令 F 为$(m \times k)$阶矩阵:

$$F = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_k(x_1) \\ f_1(x_2) & f_2(x_2) & & \\ \vdots & & \ddots & \\ f_1(x_m) & \cdots & & f_k(x_m) \end{bmatrix} \rightarrow (m \times k)$$

且 G 为$(k \times n)$阶矩阵:

$$G = \begin{bmatrix} g_1(y_1) & g_1(y_2) & \cdots & g_1(y_n) \\ g_2(y_1) & g_2(y_2) & & \\ \vdots & & \ddots & \\ g_k(y_1) & \cdots & & g_k(y_n) \end{bmatrix} \rightarrow (k \times n)$$

由式(2.7)可得出:

$$W = FG = \begin{bmatrix} \sum_{j=1}^{k} f_j(x_1) g_j(y_1) & \sum_{j=1}^{k} f_j(x_1) g_j(y_2) & \cdots & \sum_{j=1}^{k} f_j(x_1) g_j(y_n) \\ \sum_{j=1}^{k} f_j(x_2) g_j(y_1) & \sum_{k=1}^{k} f_j(x_2) g_j(y_2) & & \vdots \\ \vdots & & \ddots & \\ \sum_{j=1}^{k} f_j(x_m) g_j(y_1) & \cdots & & \sum_{j=1}^{k} f_j(x_m) g_j(y_n) \end{bmatrix} \rightarrow (m \times n) \quad (2.8)$$

对于 x 和 y 的任意一种数值组合,矩阵的乘积运算都可实现级数的求和。可以看出,对于向量 x 和 y 的任何元素组合,矩阵乘积运算为级数求和提供了一种非常简便的方法。

考虑矩阵乘积的三种特殊情况:

1. 行向量和列向量乘积
2. 列向量和行向量乘积
3. 行向量和矩阵的乘积

在这三种情况的讨论中,提供了一种解决工程问题的办法,这种办法充分利用 MATLAB 的优点。例如,简洁的表达形式和矩阵的多种运算方法。

第 1 种情况:行向量和列向量的乘积

令 a 为行向量:

$$a = [a_1 \ a_2 \cdots \ a_k] \rightarrow (1 \times k)$$

① 级数形式源于给定边界条件下微分方程式的解集。

其阶数为$(1\times k)$，$\boldsymbol{b}$ 为列向量：

$$b=[b_1\ b_2\cdots\ b_k]'\rightarrow(k\times 1)$$

其阶数为$(k\times 1)$。则 $\boldsymbol{a}$ 与 $\boldsymbol{b}$ 相乘的结果为标量：

$$\boldsymbol{d}=\boldsymbol{ab}=[a_1\quad a_2\quad\cdots\quad a_k]\begin{bmatrix}b_1\\b_2\\\vdots\\b_k\end{bmatrix}=\left[\sum_{j=1}^{k}a_jb_j\right]=\sum_{j=1}^{k}a_jb_j\rightarrow(1\times 1)\tag{2.9}$$

乘积的阶数为$(1\times k)(k\times 1)\rightarrow(1\times 1)$。这种运算称为两个向量的点乘。在 MATLAB 中上述两向量矩阵相乘的表达式定义为：

```
d = a * b
```

或

```
d = dot(a, b)
```

第 2 种情况：列向量和行向量的乘积

令 $\boldsymbol{b}$ 为$(m\times 1)$阶列向量，$\boldsymbol{a}$ 为$(1\times n)$阶行向量，则乘积 $\boldsymbol{H}=\boldsymbol{ba}$ 为：

$$\boldsymbol{H}=\boldsymbol{ba}=\begin{bmatrix}b_1\\b_2\\\vdots\\b_m\end{bmatrix}[a_1\ a_2\cdots\ a_n]=\begin{bmatrix}b_1a_1 & b_1a_2 & \cdots & b_1a_n\\b_2a_1 & b_2a_2 & & \\\vdots & & \ddots & \\b_ma_1 & & \cdots & b_ma_n\end{bmatrix}\rightarrow(m\times n)\tag{2.10}$$

结果为$(m\times n)$阶矩阵，乘积的阶数由$(m\times 1)(1\times n)\rightarrow(m\times n)$给出。因此，矩阵 $\boldsymbol{H}$ 中的元素$(h_{ij}=b_ia_j)$即为 $\boldsymbol{b}$ 和 $\boldsymbol{a}$ 中所有元素乘积的组合。

例 2.8　极坐标到笛卡儿坐标的变换

再次验证图 2.2 给出的极坐标到笛卡儿坐标的变换。即：

$$x=r\cos(\theta)$$
$$y=r\sin(\theta)$$

如果有半径值向量 $\boldsymbol{r}=[r_1\ r_2\cdots\ r_m]$和角度值向量 $\boldsymbol{\theta}=[\theta_1\theta_2\cdots\ \theta_n]$，则相应的笛卡儿坐标值为[①]：

$$\boldsymbol{X}=r'*\cos(\theta)=\begin{bmatrix}r_1\\r_2\\\vdots\\r_m\end{bmatrix}[\cos\theta_1\quad\cos\theta_2\quad\cdots\quad\cos\theta_n]$$

$$=\begin{bmatrix}r_1\cos\theta_1 & r_1\cos\theta_2 & \cdots & r_1\cos\theta_n\\r_2\cos\theta_1 & r_2\cos\theta_2 & & \\\vdots & & \ddots & \\r_m\cos\theta_1 & & \cdots & r_m\cos\theta_n\end{bmatrix}\tag{2.11a}$$

和

① 这种转换可由函数 pol2cart 实现，但 pol2cart 函数仅适用于 $m=n$ 的情况。

$$Y = r' * \sin(\theta) = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} [\sin\theta_1 \quad \sin\theta_2 \quad \cdots \quad \sin\theta_n]$$

$$= \begin{bmatrix} r_1\sin\theta_1 & r_1\sin\theta_2 & \cdots & r_1\sin\theta_n \\ r_2\sin\theta_1 & r_2\sin\theta_2 & & \\ \vdots & & \ddots & \\ r_m\sin\theta_1 & & \cdots & r_m\sin\theta_n \end{bmatrix} \qquad (2.11b)$$

这样就将极坐标映射到笛卡儿坐标中去。在例 2.9 中将说明这种变换对于绘制效果图是极为有用的。

例 2.9　环型薄膜的形状

考虑固态环型薄膜的形状模型,沿外边界受压,外边界 $r = 1$:

$$z(r, \phi) = J_1(3.8316r)\cos(\phi)$$

其中,$J_1(x)$为第一类 1 阶贝塞尔函数[①],(r, ϕ)为薄膜上任意一点的极坐标值。贝塞尔函数由下式确定。

```
besselj(n, x)
```

其中,n 为阶数,x 为参数。薄膜 r 中心处为坐标原点,薄膜的固有频率参数为 3.8316。该模型外形可由下面的外形绘制函数画出:

```
mesh(x, y, z)
```

其中,(x, y)为表面函数 $z(x, y)$上一个点的平面坐标。mesh 函数将在 7.2 节详细讨论。以 $\Delta r = 0.05$ 和 $\theta = \pi/20$ 为增量在 $0 \leqslant r \leqslant 1$ 和 $0 \leqslant \theta \leqslant 2\pi$ 区间内利用式(2.11)绘制表面图,程序如下:

```
r=[0:0.05:1]';                         %(21×1)
phi=0:pi/20:2*pi;                      %(1×41)
x=r*cos(phi);                          %(21×41)
y=r*sin(phi);                          %(21×41)
z=besselj(1,3.8316*r)cos(phi);         %(21×41)
mesh(x,y,z)
```

因为要在笛卡儿坐标系中绘出外表面图,故需要进行坐标变换。由于 cos,sin 和 besselj 函数允许向量作为输入参数,并且返回相同阶次的向量,所以这个过程是可实现的。程序执行的结果如图 2.4 所示。

例 2.10　拉普拉斯方程求解

在 $u(0, \xi) = u(1, \xi) = u(\eta, \infty) = 0, u(\eta, 0) = \eta(1-\eta)$边界条件下,拉普拉斯方程的解为:

---

① 参见 F. B. Hildebrand, *Advanced Calculus for Applications*, Prentice Hall, Upper Saddle River, NJ, 1976 年。

$$u(\eta,\xi) = 4\sum_{n=1}^{N\to\infty}\frac{1-\cos(n\pi)}{(n\pi)^3}e^{-n\pi\xi}\sin(n\pi\eta)$$

其中,$0\leqslant\eta\leqslant 1,\xi\geqslant 0$。当 $N=25$,增量 $\Delta\eta=0.025,\Delta\xi=0.05$ 直到 $\xi_{\max}=0.7$ 时,利用函数 mesh(η, ξ, $u(\xi,\eta)$)绘制 $u(\xi,\eta)$表面图。

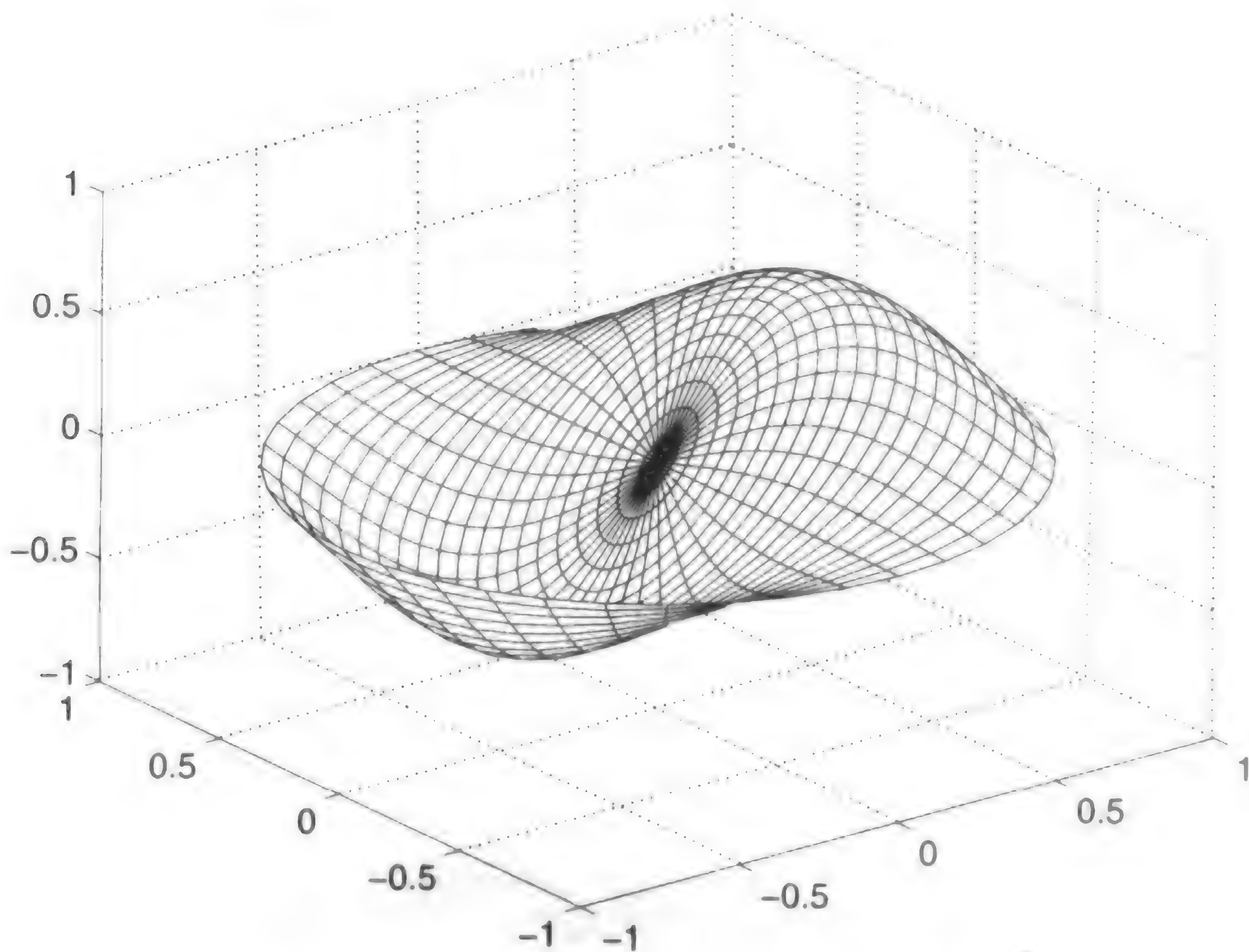


图 2.4　一个受压的固态环形薄膜外形

级数表达式的计算可通过处理构成和值的乘积项实现,即通过处理使得和值中的表达式满足式(2.8)和式(2.10)的条件。处理步骤如下:首先,设 n 为$(1\times N)$阶向量,η 为$(1\times N_e)$阶向量,ξ 为$(1\times N_x)$阶向量。然后,将阶次置于和值对应项之下,即:

$$\underbrace{\left(\frac{1-\cos n\pi}{(n\pi)^3}\right)}_{1\times N}\times\underbrace{(e^{-n\pi\xi})}_{(1\times N)(1\times N_x)}\times\underbrace{(\sin-n\pi\eta)}_{(1\times N)(1\times N_e)}$$

式中各向量长度不能满足点乘计算要求。其中第 2 项和第 3 项的计算结果不正确,为此,应对$(1\times N)$阵列进行转置操作,从而使得第 2 项计算结果的阶次为:$(1\times N)'\ (1\times N_x)\to(N\times N_x)$,第 3 项计算结果的阶次为$(1\times N)'\ (1\times N_e)\to(N\times N_e)$,表达式中第 1 项不需要进行矩阵的乘运算,只进行指数的点运算和除运算。但因 MATLAB 表达式遵循从左到右的计算规则,因此如果令其阶次与第 2 项相同,即可实现与第 2 项的点乘运算。故应利用 meshgrid 函数对第 1 项进行转换操作,使其由$(1\times N)$阶向量转换为与第 2 项相同的$(N\times N_x)$阵列。转换完成后,可对修正后的第 1 项和第 2 项进行点乘操作,结果记为 R_{nx},为$(N\times N_x)$阶矩阵。接下来应将 R_{nx} 和修正后阶次为$(N\times N_e)$的第 3 项相乘。计算前,应对 R_{nx} 进行转置操作以使乘积的中间维数相等,即:$(R_{nx})'(N\times N_e)\to(N\times N_x)'(N\times N_e)\to(N_x\times N)(N\times N_e)\to(N_x\times N_e)$,实现了在 n 的取值范围内对 ξ 和 η 的所有组合进行了求和。

程序为:

```
n = (1:25) * pi;                              %(1×25)
eta = 0:0.025:1;                              %(1×41)
```

```
xi = 0:0.05:0.7;                                      %(1×15)
[X1,temp1] = meshgrid(xi,(1 - cos(n))./n.^3);         %(25×15)
tempe2 = exp( - n' * xi);                             %(25×15)
Rnx = templ. * temp2;                                 %(25×15)
tempe3 = sin(n' * eta);                               %(25×41)
u = 4 * Rnx' * temp3;                                 %(15×41)
mesh(eta,xi,u)
```

结果如图2.5所示。mesh命令中,MATLAB允许其前两个参数 ***eta*** 和 ***xi*** 为向量,向量长度与 u 的阶数统一。mesh函数的使用方法可参见帮助文件。

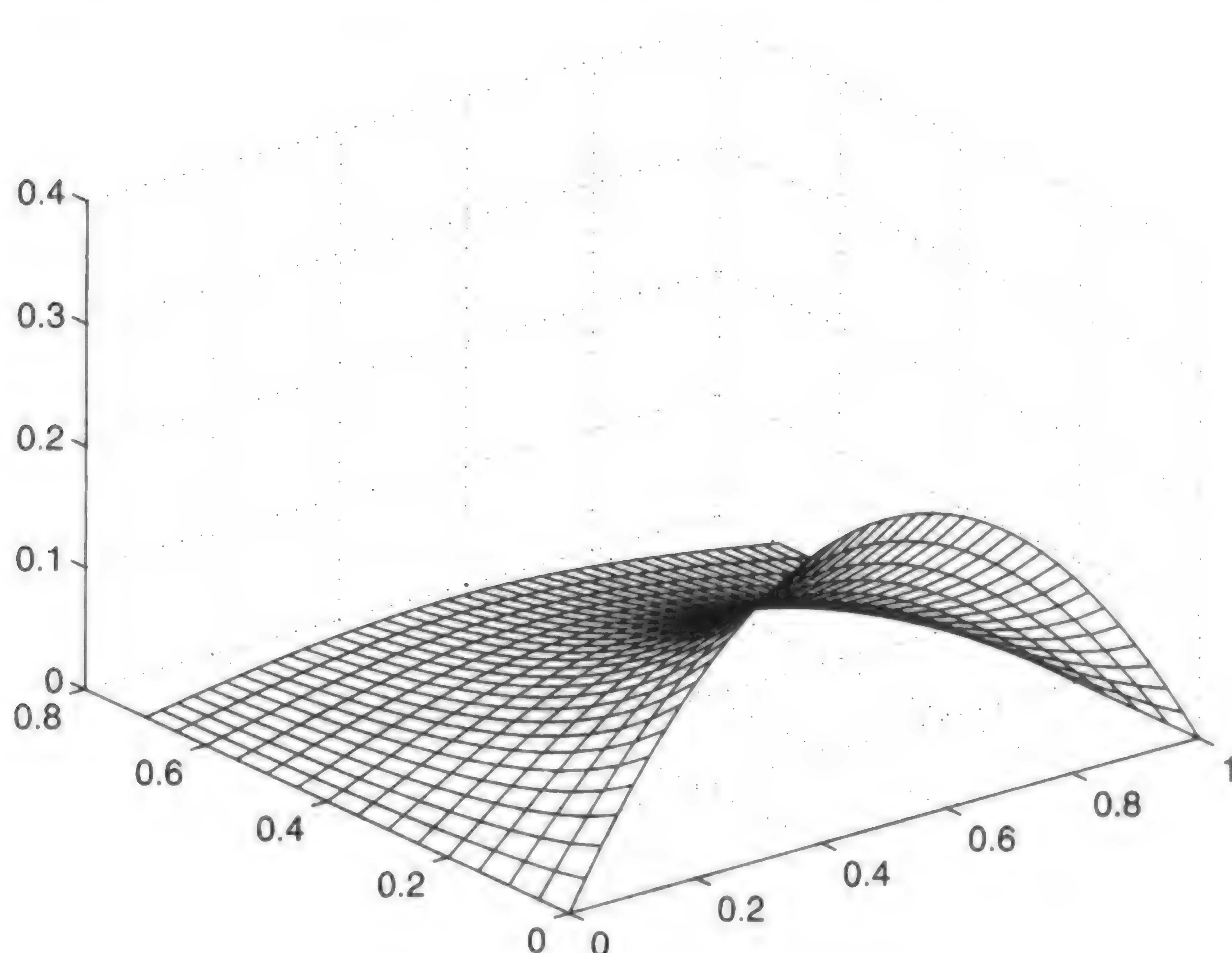


图2.5 拉普拉斯方程解集的图形显示

第3种情况:行向量和矩阵的乘积

令 $\boldsymbol{B}$ 为 $(m\times n)$ 阶矩阵且 $\boldsymbol{a}$ 为 $(1\times m)$ 阶行向量。则乘积 $\boldsymbol{g}=\boldsymbol{aB}$ 为:

$$\boldsymbol{g}=\boldsymbol{aB}=[a_1\quad a_2\quad\cdots\quad a_m]\begin{bmatrix}b_{11} & b_{12} & \cdots & b_{1n}\\ b_{21} & b_{22} & & \\ \vdots & & \ddots & \\ b_{m1} & & \cdots & b_{mn}\end{bmatrix}$$

$$=\left[\sum_{k=1}^{m}a_kb_{k1}\quad\sum_{k=1}^{m}a_kb_{k2}\quad\cdots\quad\sum_{k=1}^{m}a_kb_{kn}\right]\rightarrow(1\times n)\tag{2.12}$$

因为阶数乘积为 $(1\times m)(m\times n)\rightarrow(1\times n)$,即结果为 $(1\times n)$ 阶行向量。

结果说明如下,考虑级数①:

$$r(x)=\sum_{k=1}^{m}p_kh_k(x)\tag{2.13}$$

x 的取值可以是 x_1, x_2, $\cdots$, x_n。则 $r(x_i)$ 的表达式如下:

① 该形式级数可由特定边界条件下微分方程的解或由周期函数的傅里叶级数展开式获得。

$$r(x_i) = \sum_{j=1}^{m} p_k h_k(x_i) \quad i = 1,2,\cdots,n$$

$r(x_i)$作为$(1\times n)$阶向量 $\boldsymbol{r}$ 的一个元素。令 $\boldsymbol{p}$ 为$(1\times m)$阶向量,其元素为 p_k,$\boldsymbol{V}$ 为$(m\times n)$阶矩阵,形式如下:

$$V = \begin{bmatrix} h_1(x_1) & h_1(x_2) & \cdots & h_1(x_n) \\ h_2(x_1) & h_2(x_2) & & \\ \vdots & & \ddots & \\ h_m(x_1) & & & h_m(x_n) \end{bmatrix} \rightarrow (m\times n)$$

则 $\boldsymbol{r}=\boldsymbol{pV}$ 可表示为:

$$\boldsymbol{r} = \boldsymbol{pV} = \left[\sum_{k=1}^{m} p_k h_k(x_1) \quad \sum_{k=1}^{m} p_k h_k(x_2) \quad \cdots \quad \sum_{k=1}^{m} p_k h_k(x_n) \right] \rightarrow (1\times n)$$

例 2.11　傅里叶级数求和

脉冲宽度为 d,周期为 T 的矩形脉冲的傅里叶级数由下式给出[①]:

$$f(\tau) = \frac{d}{T}\left[1 + 2\sum_{k=1}^{K\to\infty} \frac{\sin(k\pi d/T)}{k\pi d/T}\cos(2\pi k\tau)\right]$$

其中,$\tau = t/T$。可以看出,本式是式(2.14)的一个具体表示形式。对 $f(\tau)(K=150)$的 150 个值求和,并绘制当 $d/T=0.25$, $-1/2<\tau<1/2$ 时的图形。图形的绘制可通过函数 plot(x, y) 实现。其中,$x=\tau$,$y=f(\tau)$(参见 6.2 节 plot 函数的使用说明)。

比较 $f(\tau)$的级数表达式与式(2.14)给出的表达式,有:

$$r(x_i)\rightarrow f(\tau_i)$$

$$p_k \rightarrow \frac{\sin(k\pi d/T)}{(k\pi d/T)}$$

$$h_k(x_i)\rightarrow h_k(\tau_i)\rightarrow \cos(2\pi k\tau_i)$$

如果 k 的阶次为$(1\times K)$,τ 的阶次为$(1\times N_\tau)$,则 p_k 的阶次为$(1\times K)$。可以看出,因乘积中间阶次不相等,故 $k\tau\rightarrow(1\times K)(1\times N_\tau)$的向量乘积不能实现。为满足矩阵相乘的条件,首先应对矩阵 k 进行转置操作,从而 $k'\tau\rightarrow(1\times K)'(1\times N_\tau)\rightarrow(K\times N_\tau)$,则有 $ph\rightarrow(1\times K)(K\times N_\tau)\rightarrow(1\times N_\tau)$, 即实现了在 $k=1, \cdots, K$ 范围内的求和操作。

程序为:

```
k = 1:150;                                  %(1×150)
tau = linspace(-0.5,0.5,100);               %(1×100)
sk = sin(pi*k/4)./(pi*k/4);                 %(1×150)
cntau = cos(2*pi*k'*tau);                   %(150×100)
f = 0.25*(1+2*sk*cntau);                    %(1×150)(150×100)→(1×100)
plot(tau,f)
```

程序执行结果参见图 2.6。

① 参见 H. P. Hsu, *Applied Fourier Analysis*, Harcourt Brace Jovanovich, San Diego, CA, 1984 年。

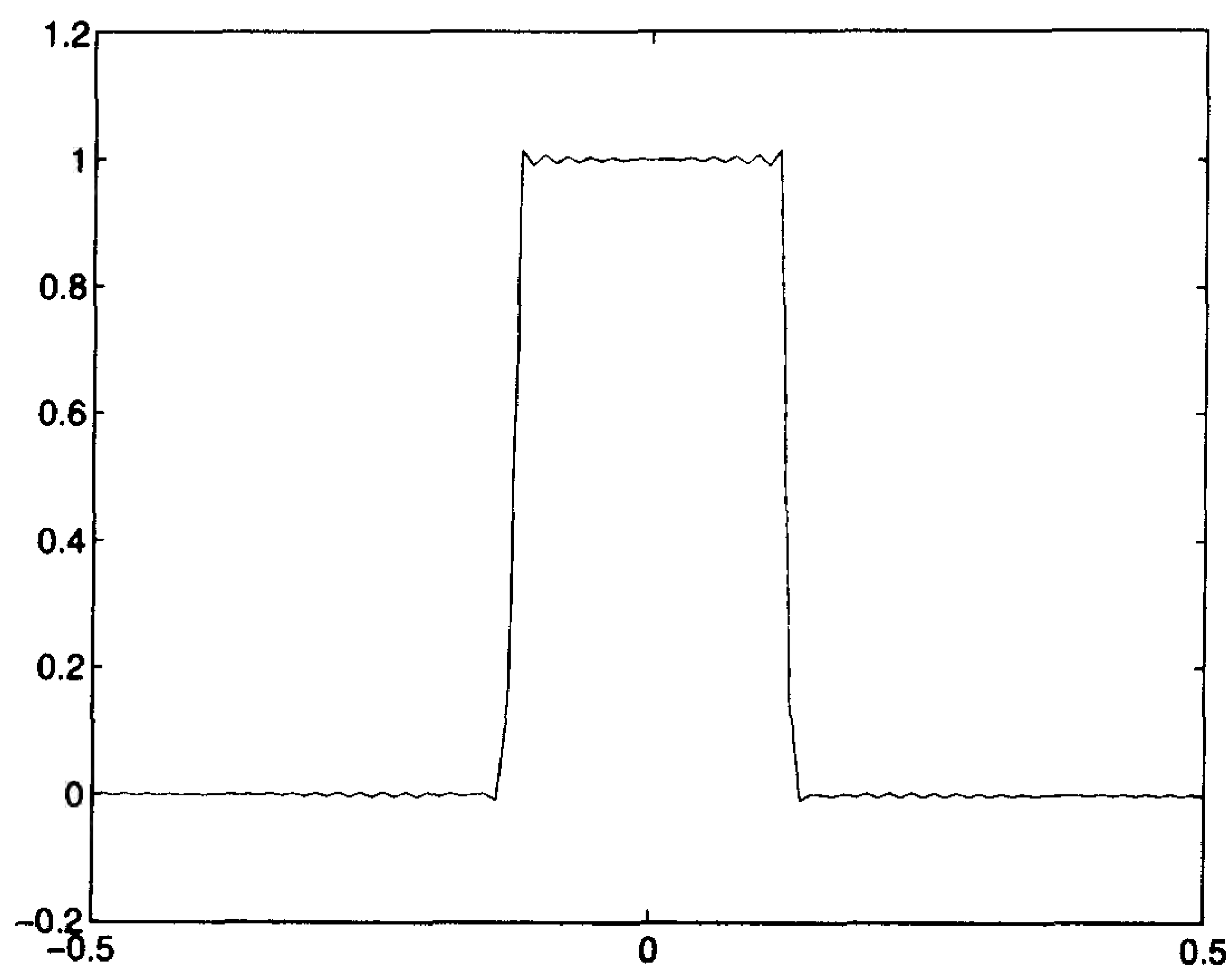


图 2.6 周期性脉冲的傅里叶级数求和

2.6.3 行列式

$(n \times n)$阶行列式A定义如下：

$$|A| = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \cdots & \\ a_{n1} & & \cdots & a_{nn} \end{vmatrix}$$

$n=2$ 时，

$$|A| = a_{11}a_{22} - a_{12}a_{21}$$

$n=3$ 时，

$$|A| = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}$$

行列式的 MATLAB 表达式为：

```
det(a)
```

例如，如果 A 定义为：

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

则：

```
A=[1, 3;4, 2];
d=det(A)
```

执行结果为：

```
d=
  -10
```

行列式的特征值表达式为：

$$|A - \lambda B| = 0$$

其中，A 和 B 是 $(n \times n)$ 阶矩阵，$\lambda_j (j = 1, 2, \cdots, n)$ 为方程的根（或特征值）。第 9 章介绍了该类问题在振动领域中的应用。多项式方程的解可由函数 eig 得出，其应用形式之一为：

```
lambda = eig(A, B)
```

例 2.12　弹簧质量系统特征值

具有 3 个自由度的弹簧质量系统特征方程为：

$$|K - \omega^2 M| = 0$$

其中，刚度矩阵 K 为：

$$K = \begin{bmatrix} 50 & -30 & 0 \\ -30 & 70 & -40 \\ 0 & -40 & 50 \end{bmatrix}$$

质量矩阵为：

$$M = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1.4 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

特征值 $\lambda = \omega^2$ 与系统固有频率的关系式为 $\omega_j = \sqrt{\lambda_j}$，其中 $j = 1, 2, 3$。

固有频率可由下列程序得出：

```
K = [50, -30, 0; -30, 70, -40;0, -40, 50];
M = diag([3, 1.4, 5]);
w = sqrt(eig(K, M))
```

执行结果为：

```
w =
  1.6734
  3.7772
  7.7201
```

例 2.13　多项式变换

有以下形式的多项式：

$$ax^2 + by^2 + cz^2 + 2dxy + 2exz + 2gyz$$

其中，a，b，c，d，e 和 g 为实数。多项式可变换为实对角阵形式：

$$r_1 x'^2 + r_2 y'^2 + r_3 z'^2$$

其中，x'，y' 和 z' 为另一坐标系，坐标系原点也在 $(0, 0, 0)$，$r_1 \geq r_2 \geq r_3$ 为行列式

$$|A - rI| = 0$$

的根，I 为单位阵，A 为实对称阵：

$$A = \begin{bmatrix} a & d & e \\ d & b & g \\ e & g & c \end{bmatrix}$$

考虑多项式：

$$4x^2+3y^2-z^2-12xy+4exz-8gyz$$

则:

$$A=\begin{bmatrix}4 & -6 & 2\\ -6 & 3 & -4\\ 2 & -4 & -1\end{bmatrix}$$

为求取根 r_j,可输入:

```
r=eig([4,-6,2;-6,3,-4;2,-4,-1],eye(3))
```

执行结果为:

```
r=
  -4.0000
  -1.0000
  11.0000
```

注意,这些根不是按给定顺序排列的。如果想按预期给定的顺序排列,可利用 sort 函数。但 sort 函数仅按由最小负值到最大正值的升序排列,因此,可利用下列表达式之一得到降序排列的数值,这些表达式可适用于正实数与负实数的任意组合。

形式 #1:

```
r=eig([4,-6,2;-6,3,-4;2,-4,-1],eye(3));
r=-sort(-r)
```

执行结果为:

```
r=
  11.0000
  -1.0000
  -4.0000
```

形式 #2:

```
r=eig([4,-6,2;-6,3,-4;2,-4,-1],eye(3));
r=flipud(sort(r))
```

也可以给出上述结果值。如果 r 为行向量,则可将 flipud 更换为 fliplr。

上述语句可简化为:

```
r=-sort(-eig([4,-6,2;-6,3,-4;2,-4,-1]))
```

因此,实对角形式为:

$$11x'^2-y'^2-4z'^2$$

2.6.4 矩阵的逆

方阵 A 的逆运算:

$$A^{-1}A=AA^{-1}=I$$

上式表明 A 为非奇异阵,也就是说,其行列式不等于 $0(|A|\neq 0)$。I 为单位阵,上标"-1"代表逆。获得矩阵 A 的逆的表达式为:

```
inv(A)
```

或

```
A^-1
```

应注意 $1/A \neq A\hat{}(-1)$，$1/A$ 将产生错误。逆也可由 2.6.5 节中介绍的反斜线运算符求得。

例 2.14　矩阵的逆

考虑函数 magic 创建的(3×3)阶矩阵 M，它的逆可由下列语句获得：

```
invM = inv(magic(3))
```

执行结果为：

```
invM =
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
```

修改上述程序代码可验证方阵的乘积和它的逆都为单位阵：

```
invM = inv(magic(3));
IdentMat = invM * magic(3)
```

执行结果为：

```
IndentMat =
    1.0000         0   -0.0000
         0    1.0000         0
         0    0.0000    1.0000
```

考虑矩阵：

$$C = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 9 & 13 \\ 6 & 12 & 18 \end{bmatrix}$$

求矩阵 C 的逆可输入以下代码：

```
C = [1, 2, 3;6, 9, 13;6, 12, 18];
Iv = inv(C)
```

执行结果为：

```
Warning: Matrix is close to singular or badly scaled.
    Results may be inaccurate. RCOND = 2.176908e-018.
Iv =
  1.0e+015*
    0.0000   -0.2502    1.5012
   -0.0000    1.2510   -7.5060
         0   -0.7506    4.5036
```

结果包含错误信息，且逆矩阵一列元素为 0。第 2 行的 1.0e+015* 表示后面的每一个数都乘以 10^{15}。RCOND 为矩阵的条件数，良性矩阵的条件数接近于 1，而病态矩阵的条件数

接近于0。如果上式应用了函数 det,则 $|C| = 0$,即矩阵没有逆矩阵。

判断是否存在逆矩阵的另一个方法为:

```
rank
```

由此给出矩阵所有线性无关的行或列数。因此,对于$(n\times n)$阶矩阵,线性无关的行或列数为 n 减去矩阵的秩。本例中,rank(C)的返回值为2,表明矩阵中有 $3-2=1$ 行或列与其他行或列存在线性比例关系。本例中,可以看出第3行各元素值为第1行各对应元素值的6倍。

2.6.5 方程组求解

考虑下列 n 阶方程组,n 个未知数 x_k,$k=1, 2, \cdots, n$:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

方程组的矩阵形式可以写为:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

其中,$\boldsymbol{A}$ 为$(n\times n)$阶矩阵:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix} \rightarrow (n\times n)$$

且 $\boldsymbol{x}$ 和 $\boldsymbol{b}$ 分别为$(n\times 1)$阶向量:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \rightarrow (n\times 1) \quad 和 \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \rightarrow (n\times 1)$$

其方程的解可在矩阵方程两边同乘以 $\boldsymbol{A}^{-1}$ 求出。即:

$$\boldsymbol{A}^{-1}\boldsymbol{Ax} = \boldsymbol{A}^{-1}\boldsymbol{b}$$
$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

因为 $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$,且 $\boldsymbol{Ix} = \boldsymbol{x}$。推荐采用以下表达式求解此方程组[①]:

```
x = A \ b
```

其中,反斜线运算符代表矩阵的除法,在 MATLAB 中称之为矩阵的左除。与以下公式相比,左除具有更好的数值稳定性:

---

① 表示形式 $\boldsymbol{A}\backslash\boldsymbol{b}$ 也适用于 $\boldsymbol{A}$ 不是方阵时的情形,而 inv(A)却仅适用于 $\boldsymbol{A}$ 为方阵的情形。也就是说,如果 $\boldsymbol{A}$ 是一个$(m\times n)$阶矩阵,$\boldsymbol{x}$ 是一个$(n\times 1)$阶向量,$\boldsymbol{b}$ 是一个$(m\times 1)$阶向量,则 $\boldsymbol{Ax}=\boldsymbol{b}$,左除以 $\boldsymbol{A}\backslash\boldsymbol{b}$ 得出 $\boldsymbol{x}=\boldsymbol{cb}$,其中 $\boldsymbol{c}=(\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}'$,为矩阵 $\boldsymbol{A}$ 的准逆形式。

```
x = A^-1 * b
```

或

```
x = inv(A) * b
```

与反斜线运算符相比较,后两种方法运算速度较慢。

例 2.15 方程组求解

考虑下列方程组:

$$
\begin{aligned}
8x_1 + x_2 + 6x_3 &= 7.5 \\
3x_1 + 5x_2 + 7x_3 &= 4 \\
4x_1 + 9x_2 + 2x_3 &= 12
\end{aligned}
$$

其矩阵形式为:

$$
\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7.5 \\ 4 \\ 12 \end{bmatrix}
$$

通过以下程序段求解方程组:

```
A = [8, 1, 6;3, 5, 7;4, 9, 2];
b = [7.5, 4, 12]';
x = A \ b
```

结果为:

```
x =
     1.2931
     0.8972
    -0.6236
```

程序可简化为:

```
x = [8, 1, 6;3, 5, 7;4, 9, 2] \ [7.5, 4, 12]'
```

因为 $\boldsymbol{b} = \boldsymbol{Ax}$,所以可用下列修正的程序代码检验结果的正确性:

```
A = [8, 1, 6;3, 5, 7;4, 9, 2];
b = [7.5, 4, 12]';
x = A \ b;
z = A * x
```

结果为:

```
z =
   7.5000
   4.0000
   12.0000
```

例 2.16 固定平板的静态变形

一正方形平板,四边固定,表面载荷均匀。求解其静态变形时,首先应从以下无限方程组

中，通过舍去一些位获得常数 E_m 值[①]：

$$a_iE_i + \sum_{m=1,3,\cdots} b_{im}E_m = c_i \quad i = 1,3,5,\cdots$$

其中，

$$a_i = \frac{1}{i}\left(\tanh\alpha_i + \frac{\alpha_i}{\cosh^2\alpha_i}\right)$$

$$b_{im} = 8i\left(\pi m^3\left(1 + \frac{i^2}{m^2}\right)^2\right)^{-1}$$

$$c_i = \frac{4}{\pi^3 i^4}\left(\frac{\alpha_i}{\cosh^2\alpha_i} - \tanh\alpha_i\right)$$

且 $\alpha_i = i\pi/2$。如果仅取方程组的前 4 个，则：

$$\begin{bmatrix} a_1 + b_{11} & b_{13} & b_{15} & b_{17} \\ b_{31} & a_3 + b_{33} & b_{35} & b_{37} \\ b_{51} & b_{53} & a_5 + b_{55} & b_{57} \\ b_{71} & b_{73} & b_{75} & a_7 + b_{77} \end{bmatrix}\begin{bmatrix} E_1 \\ E_3 \\ E_5 \\ E_7 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_3 \\ c_5 \\ c_7 \end{bmatrix}$$

可以看出，系数 b_{im} 为索引 $\boldsymbol{i}$ 和 $\boldsymbol{m}$ 的函数，$\boldsymbol{i}$ 和 $\boldsymbol{m}$ 都是长度为 4 的向量。因此，可以用函数 meshgrid 将其转换为(4×4)阶矩阵以应用点运算。

可用下列程序求这 4 个方程的解：

```
m = 1:2:7;i = m;                                        %(1×4)
alp = m * pi/2;                                         %(1×4)
ai = (tanh(alp) + alp./cosh(alp).^2)./i;                %(1×4)
ci = 4. * (alp./(cosh(alp).^2) - tanh(alp))./((pi^3) * i.^4);  %(1×4)
[ii,mm] = meshgrid(i,m);                                %(4×4)
bim = (8/pi) * ii./(((1 + (ii.^2)./(mm.^2)).^2). * mm.^3);      %(4×4)
ee = (diag(ai) + bim) \ ci'
```

结果生成：

```
ee =
  -0.0480
   0.0049
   0.0023
   0.0011
```

则 E_1 = ee(1, 1) = −0.0480；E_3 = ee(2, 1) = 0.0049；E_5 = ee(3, 1) = 0.0023；E_7 = ee(4, 1) = 0.0011。

2.7 应用函数一览

本章的应用函数一览参见表 2.1。此外，本章应用到的特殊符号(如冒号、反斜线、撇号、分号、点号、圆括号、方括号和逗号)可参见表 1.7。

① 参见 S. Timoshenko 和 S. Woinowsky-Krieger, *Theory of Plates and Shells*, McGraw-Hill, New York, 1959 年，第 197~202 页。

表2.1　第2章MATLAB函数介绍

MATLAB函数	说明
besselj	第一类贝塞尔函数
cumsum	阵列累积和
det	方阵行列式
diag	方阵的对角线;创建对角阵
dot	两向量的点积
eig	矩阵特征方程的特征值和特征向量
end	阵列最后项的指针(参见表4.2)
eye	创建单位阵
find	满足逻辑表达式的阵列索引和阵列值
fliplr	从左到右交换阵列元素值
flipud	从底到顶交换阵列元素值
inv	方阵的逆
length	向量长度
linspace	创建向量的等分元素
logspace	以 $\log_{10}$ 为标度创建向量的等分元素
magic	创建任意行和列的和值相等的方阵
max	确定阵列中的最大值
mesh	生成线框几何面
meshgrid	将两个不同的向量转变为具有相同长度的阵列
min	确定阵列中的最小值
ones	生成所有元素值全为1的阵列
plot	应用线性坐标绘制平面曲线
rank	确定矩阵线性无关的行或列值
repmat	复制阵列
size	阵列的阶次或长度
sort	按升序排列阵列元素
sum	计算阵列元素的和值
zeros	创建所有元素全为0的阵列

练习

2.1　建立两个向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$,向量 $\boldsymbol{a}$ 的元素为 $2n-1$,向量 $\boldsymbol{b}$ 的元素为 $2n+1$, $n=0,1,\cdots,7$。

(a) 求 $\boldsymbol{a}$ 与 $\boldsymbol{b}$ 的和。

(b) $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的区别是什么?

(c) 求 $\boldsymbol{a}'\boldsymbol{b}$ 的乘积及其行列式的值。

(d) 求 $\boldsymbol{ab}'$ 的乘积。

2.2 给出向量 $\boldsymbol{x}=[17,-3,-47,5,29,-37,51,-7,19]$。编程使 $\boldsymbol{x}$ 按照向量 $\boldsymbol{y}=[-3,-7,-37,-47,51,29,19,17,5]$ 的形式重新排列。程序应适用于任意长度的向量。对所有向量,数值 0 与负数放在一起。也就是说,如果 0 为向量的一个元素,则将作为 y 向量的第 1 个元素。

2.3 令向量 $y=[0,-0.2,0.4,-0.6,0.8,-1.0,-1.2,-1.4,1.6]$。如果 $z=\sin(y)$,则:

(a) 求向量 z 中全部负数值的最大值和最小值。

(b) 求向量 z 中全部正数值的平方根。

2.4 (a) 按对数坐标创建 8 等分向量,向量第 1 个元素的值为 6,最后一个元素的值为 106。

(b) 显示(a)中所创建向量的第 5 个元素值。

(c) 创建一个向量,令其元素为(a)中所创建向量的第 1 个、第 3 个、第 5 个和第 7 个元素。

2.5 令 z = magic(5)。

(a) 按以下给定的顺序执行对 z 的运算:

(i) 第 2 列除以$\sqrt{3}$。

(ii) 将第 3 行元素加到第 5 行(第 3 行保持不变)。

(iii) 将第 1 列与第 4 列相乘的结果放在第 1 列。

(iv) 令对角元素为 2。

(b) 如果将(a)中求出的结果记为 q,则显示 qq' 的对角元素值。
[答案:[486 104189 7300 44522 111024]'。]

(c) 显示 q 中每一元素的平方。

(d) 显示(c)中各元素的最大值和最小值。

2.6 令 $\boldsymbol{w}$ = magic(2),$\boldsymbol{w}'$ 为 $\boldsymbol{w}$ 的转置。

(a) 利用函数 repmat,创建下列(4×4)阶矩阵:

$$\begin{bmatrix} \boldsymbol{w} & \boldsymbol{w} \\ \boldsymbol{w} & \boldsymbol{w} \end{bmatrix}$$

(b) 利用函数 repmat,创建下列(6×2)阶矩阵:

$$\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w} \\ \boldsymbol{w} \end{bmatrix}$$

(c) 利用函数 repmat 和列参数过程创建下列(6×4)阶矩阵:

$$\begin{bmatrix} \boldsymbol{w} & \boldsymbol{w}' \\ \boldsymbol{w} & \boldsymbol{w}' \\ \boldsymbol{w} & \boldsymbol{w}' \end{bmatrix}$$

(d) 不采用 repmat 函数,仅用列参数过程和行参数重复过程(a),(b)和(c)。

2.7 令 $\boldsymbol{x}$ = magic(3)。

(a) 创建新矩阵,令 $\boldsymbol{x}$ 中的每一行都向上移动一行,第 1 行变为最后一行。

(b) 创建新矩阵,令 x 中的每一列都向右移动一列,最后一列变为第 1 列。

2.8 曲柄滑块机构如图 2.7 所示,滑块的位移为:

$$s = a\cos(\varphi) + \sqrt{b^2 - (a\sin(\varphi) - e)^2}$$

位移 s 是角度 φ(以角度表示)的函数,当 $a = 1, b = 1.5, e = 0.3, 0 \leqslant \varphi \leqslant 360°$ 时,使用函数 plot(φ,s)绘制 s 的图形。提示:三角函数参数以弧度表示。

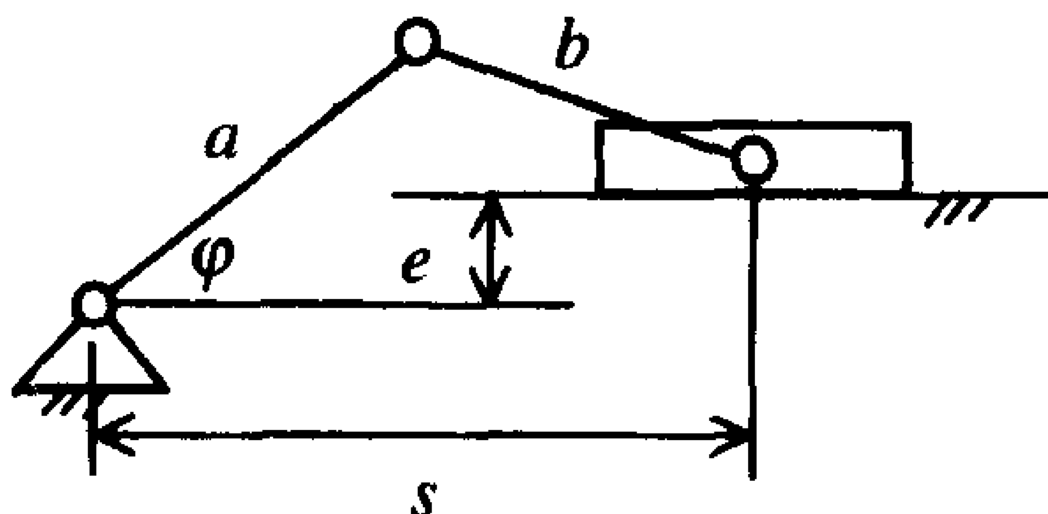


图 2.7　曲柄滑块机构

2.9 在表示周期性矩形脉冲的级数中,将总功率的百分数 P 作为 N_H 的函数,其级数展开式为:

$$P = 100P_o/P_T\%$$

其中,P_T 为信号中的总功率:

$$P_o = 1 + 2\sum_{n=1}^{N_H} \frac{\sin^2(n\pi\tau_o/T)}{(n\pi\tau_o/T)^2}$$

τ_0/T 为脉冲宽度相对于周期的比值。若 $\tau_o/T = 1/\sqrt{19}$,则 $P_T \cong 4.3589$。在 $2 \leqslant N_H \leqslant 25$ 的区间内利用函数 plot(N_H, P_o)绘制 N_H 的图形。

2.10 考虑下列乘积[①]:

$$S_N = \prod_{n=1}^{N}\left(1 - \frac{x^2}{n^2 - a^2}\right)$$

当 $N \to \infty$ 时,

$$S_\infty = \frac{a}{\sin\pi a\sqrt{a^2 + x^2}}\sin\left(\pi\sqrt{a^2 + x^2}\right)$$

相对误差定义为:

$$e_N = 100\frac{S_N - S_\infty}{S_\infty}\%$$

如果 x 在 1 到 5 之间以 0.5 的增量变化,且 $a = \sqrt{2.8}$,则 $N = 100$ 时在 x 的 9 个取值处相对误差为多少?使用 prod 函数计算 S_N。

[答案:e_{100} = [1.0001 2.2643 4.0610 6.4176 9.3707 12.9670 17.2642 22.3330 28.2588]。]

2.11 在韦布尔概率密度函数(参见 14.2.2 节)中,参数 δ 的估计值可由下式获得:

$$\delta = \left[\frac{1}{n}\sum_{i=1}^{n} x_i^\beta\right]^{1/\beta}$$

① 参见 L. B. W. Jolley,出处同上。

其中,x_i 为 n 范围内的某一值,β 为另一已知参数。如果 x = [72, 82, 97, 103, 113, 117, 126, 127, 127, 139, 154, 159, 199, 207],且 $\beta = 3.644$,求取 δ 值。

2.12 从球面坐标到笛卡儿坐标的转换公式为:

$$x = b\sin\phi\cos\theta$$
$$y = b\sin\phi\sin\theta$$
$$z = b\cos\phi$$

ϕ 在 $0 \leqslant \phi \leqslant 90°$的范围之内 10 等分,$\theta$ 在 $0 \leqslant \theta \leqslant 360°$的范围之内 24 等分。当 $b = 2$ 时,用 mesh(x, y, z)函数绘出这个半球形。

2.13 在 $0.1 \leqslant x \leqslant 1$ 的范围内将 x 值 5 等分,且当 $N = 25$ 时估算下列级数值($N \to \infty$):

$$\sum_{N \to -\infty}^{N \to \infty} \frac{1}{n^4 + x^4} = \frac{2\pi^4}{y^3} \frac{\sinh y + \sin y}{\cosh y - \cos y} \quad y = \pi x\sqrt{2}$$

并将这些值与实际值进行比较。

2.14 一类 n 阶贝塞尔函数由下式给出:

$$J_n(x) = \sum_{k=0}^{K \to \infty} \frac{(-1)^k (x/2)^{2k+n}}{k!\Gamma(k+1+n)}$$

在 $1 \leqslant x \leqslant 6$ 的范围内将 x 值 6 等分,当 $n = 2, K = 25$ 时求 $J_n(x)$的向量。伽马函数 Γ 和阶乘都可由 gamma 函数获得。将求出值与用 MATLAB 内置的 besselj 函数求出值进行比较。

2.15 当 $n = 7$ 时,证明下列级数① 之和为给出值:

$$\sum_{k=1}^{2n-1} \cos(k\pi/n) = -1$$

2.16 如果,

$$X'X = I$$

则矩阵称为正交矩阵,因此,$(X'X)^{-1} = I$。证明下列矩阵为正交阵。

$$w = \frac{1}{2}\begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \quad q = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

2.17 如图 2.8 所示,考虑平面上具有三个自由度的连杆机构。点 O_3 相对于固定坐标系 O_0 的位置和方向为:

$$T_3 = A_1 A_2 A_3$$

其中,

$$A_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 & a_j\cos\theta_j \\ \sin\theta_j & \cos\theta_j & 0 & a_j\sin\theta_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad j = 1,2,3$$

且,

① 参见 L. B. W. Jolley 第 86 页~第 87 页,出处同上。

$$T_3 = \begin{bmatrix} u_x & v_x & 0 & q_x \\ u_y & v_y & 0 & q_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

元素 q_x 和 q_y 为 O_3 点相对于坐标原点 O_0 的 (x, y) 坐标。如果 $\theta_j = 30°, j = 1, 2, 3$ 且 $a_1 = 1, a_2 = 2, a_3 = 3$,求 O_3 点相对于坐标原点 O_0 的位置,并求出 (x_3, y_3) 轴系的方向。

[答案:$q_x = 1.8660, q_y = 5.2321, x_3$ 平行于 y_0, y_3 平行于 x_0,但其方向相反。]

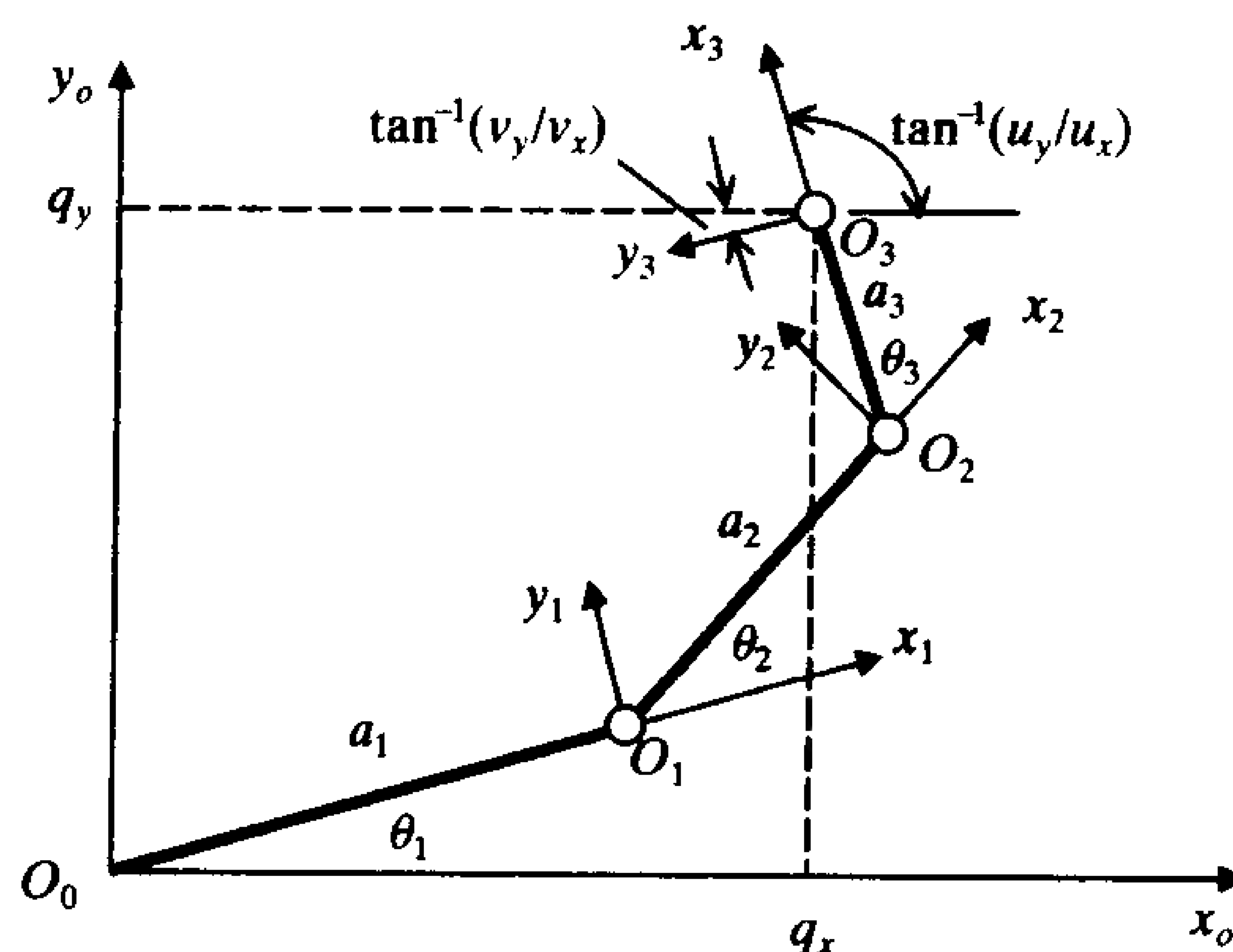


图 2.8　平面上具有三个自由度的连杆机构

2.18　在多重线性回归分析中,使用下列矩阵(参见练习 14.13):

$$H = X(X'X)^{-1}X'$$

如果,

$$X = \begin{bmatrix} 17 & 31 & 5 \\ 6 & 5 & 4 \\ 19 & 28 & 9 \\ 12 & 11 & 10 \end{bmatrix}$$

求对角阵 H。

[答案:对角阵 $H = [0.7294\ 0.9041\ 0.4477\ 0.9188]'$。]

2.19　在给定的 τ 值在 200 以下范围内使用函数 plot(τ, $f(\tau)$)绘出下述级数[①] 的图形。除非特别指明,所有级数仅求出前 200 项的和,用 2.6.2 节中第 3 种情况给出的向量相乘进行求和运算。

(a) 方波:

$$f(\tau) = \frac{4}{\pi}\sum_{n=1,3,5,\cdots}\frac{1}{n}\sin(2n\pi\tau) \qquad -\frac{1}{2} \leqslant \tau \leqslant \frac{1}{2}$$

(b) 锯齿波:

① 参见 H. P. Hsu,出处同上。

$$f(\tau)=\frac{1}{2}+\frac{1}{\pi}\sum_{n=1}\frac{1}{n}\sin(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(c) 锯齿波:

$$f(\tau)=\frac{1}{2}-\frac{1}{\pi}\sum_{n=1}\frac{1}{n}\sin(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(d) 三角波:

$$f(\tau)=\frac{\pi}{2}-\frac{4}{\pi}\sum_{n=1}\frac{1}{(2n-1)^2}\cos((2n-1)\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(e) 正弦整流波:

$$f(\tau)=\frac{2}{\pi}+\frac{4}{\pi}\sum_{n=1}\frac{1}{1-4n^2}\cos(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(f) 半正弦波:

$$f(\tau)=\frac{1}{\pi}+\frac{1}{2}\sin\pi\tau-\frac{2}{\pi}\sum_{n=2,4,6,\cdots}^{106}\frac{\cos n\pi\tau}{n^2-1}\quad -2\leqslant\tau\leqslant 2$$

(g) 指数:

$$f(\tau)=\frac{e^{2\pi}-1}{\pi}\left[\frac{1}{2}+\sum_{n=1}^{250}\frac{1}{1+n^2}(\cos n\tau-n\sin n\tau)\right]\quad 0\leqslant\tau\leqslant 4\pi$$

使用 350 个 τ 值显示结果。

(h) 梯形波:

$$f(\tau)=\frac{4}{a^2}\sum_{n=1,3,5,\cdots}^{105}\frac{\sin n\pi a}{(\pi n)^2}\sin n\pi\tau\quad -2\leqslant\tau\leqslant 2$$

令 $a=0.25$。

2.20 考虑两个级数①:

$$S_{1N}=\sum_{n=1}^{N}\frac{\cos(n\theta)}{n^2+a^2}\qquad 0<\theta<\pi$$

$$S_{2N}=\sum_{n=1}^{N}\frac{n\sin(n\theta)}{n^2+a^2}\qquad 0<\theta<2\pi$$

其中,当 $N\to\infty$ 时:

$$S_{1\infty}=\frac{\pi\cosh[a(\pi-\theta)]}{2a\sinh\pi a}-\frac{1}{2a^2}\qquad 0<\theta<\pi$$

$$S_{2\infty}=\frac{\pi\sinh[a(\pi-\theta)]}{2\sinh\pi a}\qquad 0<\theta<2\pi$$

相对误差定义为:

$$e_{jN}=100\,\frac{S_{jN}-S_{j\infty}}{S_{j\infty}}\%\qquad j=1,2$$

如果 θ 在 10°~80°的范围内以 10°的增量变化,且 $a=\sqrt{3}$,则当 $N=25$ 时,求这两个级数所对应 8 个 θ 值的相对误差。用 2.6.2 节中第 3 种情况给出的向量相乘进行计算。

[答案:e_1 = [-1.2435 0.8565 0.8728 -1.9417 -0.9579 -8.1206 0.7239 1.1661],

① 参见 L. B. W. Jolley,出处同上。

$e_2 = [8.0538\ 10.4192\ -8.9135\ -5.4994\ 12.9734\ -0.5090\ -17.2259\ 11.2961]$。]

2.21　边界 $\eta = 1$ 处为恒温的矩形平板，其无量纲稳态温度分布函数为①：

$$T(\eta,\xi) = \frac{4}{\pi}\sum_{n=1,3,5}^{\infty}\frac{\sinh(n\pi\alpha\eta)}{n\sinh(n\pi\alpha)}\sin(n\pi\xi)$$

其中，$\eta = x/d$，$\xi = y/b$，d 和 b 分别为平板在 x 和 y 方向的长度，$\alpha = d/b$，$0 \leqslant \eta \leqslant 1$ 且 $0 \leqslant \xi \leqslant 1$。令 $\Delta\eta = \Delta\xi = 1/14$，当 $\alpha = 2$ 时，用 mesh(ξ, η, T)函数显示整个平板的温度分布。用 2.6.2 节中第 2 种情况给出的向量相乘进行计算。

2.22　在一个初始速度为 0 的弦中，初始位移如下：

$$u(\eta,0) = \frac{\eta}{a} \qquad 0 \leqslant \eta \leqslant a$$

$$u(\eta,0) = \frac{1-\eta}{1-a} \quad a \leqslant \eta \leqslant 1$$

波的直线传播位移公式由下式给出：

$$u(\eta,\tau) = \frac{2}{a\pi(1-a)}\sum_{n=1}^{N\to\infty}\frac{\sin n\pi a}{n^2}\sin(n\pi\eta)\cos(n\pi\tau)$$

$N = 50$，$a = 0.25$，$\Delta\eta = 0.05$，$0 \leqslant \tau \leqslant 2$ 时，用函数 mesh(τ, η, u)显示 $u(\eta, \tau)$，用 2.6.2 节中第 2 种情况给出的向量相乘进行计算。结果如图 2.9 所示，是通过使用图形窗口中的"旋转"图标得到的。

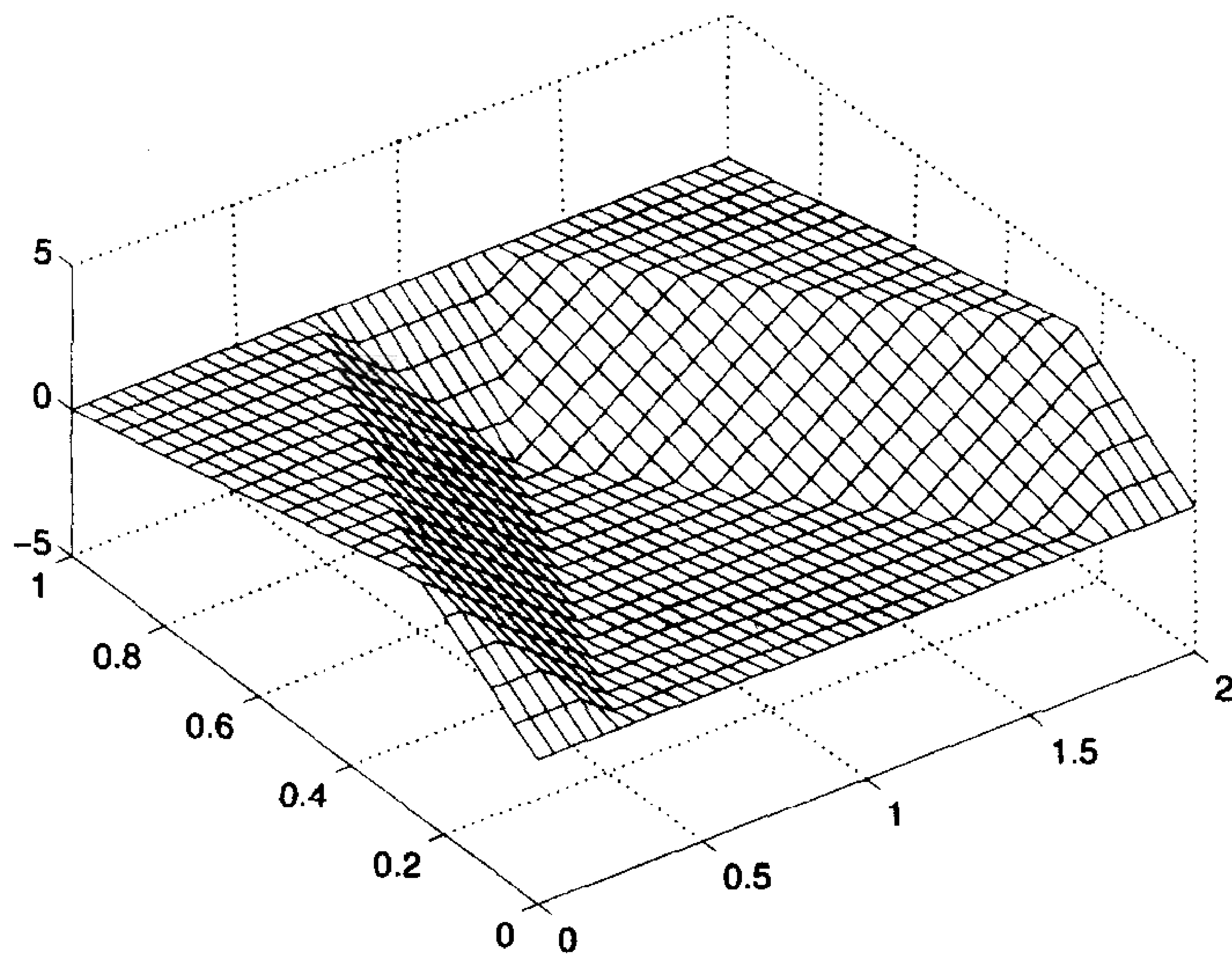


图 2.9　在一根弦中初始位移的传播

2.23　给出两个矩阵：

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 4 \\ 31 & 67 & 9 \\ 7 & 5 & 9 \end{bmatrix} \quad 和 \quad \boldsymbol{B} = \begin{bmatrix} 11 & 34 & 6 \\ 7 & 13 & 8 \\ 43 & 10 & 53 \end{bmatrix}$$

① 参见 H. P. Hsu，出处同上。

证明$|AB| = |A||B|$。

2.24 给出以下方程组：

$$\begin{aligned} 16s + 32u + 33p + 13w &= 91 \\ 5s + 11u + 10p + 8w &= 16 \\ 9s + 7u + 6p + 12w &= 5 \\ 34s + 14u + 15p + w &= 43 \end{aligned}$$

求 s, u, p 和 w 的值，并求出系数的行列式。

[答案：$s = -0.1258, u = -8.7133, p = 11.2875, w = -0.0500$。行列式 = 7680。]

2.25 考虑两个由不同物质构成的长圆筒，其中一个圆筒正好套在另一个圆筒里面。里侧圆筒的内径为 a，外径为 b；外侧圆筒的内径为 b，外径为 c。里侧圆筒的扬氏模量和泊松比分别为 E_1 和 ν_1，外侧圆筒的扬氏模量和泊松比分别为 E_2 和 ν_2。径向应力 σ_{rr}，环向应力 $\sigma_{\theta\theta}$ 和径向位移 μ_r 分别由下式给出：

$$\begin{aligned} \sigma_{rri}(r) &= \frac{A_i}{r^2} + B_i \\ \sigma_{\theta\theta i}(r) &= \frac{-A_i}{r^2} + B_i \quad i = 1,2 \\ u_{ri}(r) &= \frac{-(1+v_i)}{rE_i}A_i + \frac{(1-v_i)}{E_i}rB_i \end{aligned} \tag{a}$$

其中，$i = 1$ 表示里侧圆筒，$i = 2$ 表示外侧圆筒。

如果外侧圆筒的外表面有一个压缩的径向位移 U_o，而里侧圆筒的内表面没有径向应力，则下列 4 个边界条件可用于确定 A_i 和 B_i，$i = 1, 2$：

$$\begin{aligned} \sigma_{rr1}(a) &= 0 \\ \sigma_{rr1}(b) &= \sigma_{rr2}(b) \\ u_{r1}(b) &= u_{r2}(b) \\ u_{r2}(c) &= -U_o \end{aligned} \tag{b}$$

将式(a)代入式(b)，得到下列方程组：

$$\begin{bmatrix} 1 & a^2 & 0 & 0 \\ 1 & b^2 & -1 & -b^2 \\ -(1+v_1) & (1-v_1)b^2 & (1+v_2)E_1/E_2 & -(1-v_2)b^2E_1/E_2 \\ 0 & 0 & -(1+v_2) & (1-v_2)c^2 \end{bmatrix} \begin{Bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -U_oE_2c \end{Bmatrix}$$

当 $\nu_1 = \nu_2 = 0.4$, $E_1 = 3 \times 10^5$ psi, $E_2 = 3.5 \times 10^4$ psi, $U_o = 0.01$ in, $a = 0.192$ in, $b = 0.25$ in, $c = 0.312$ in 时，求里侧圆筒和外侧圆筒在 $r = b$ 处的环向应力。

[答案：$\sigma_{\theta\theta 1}(b) = -9571.8$ psi, $\sigma_{\theta\theta 2}(b) = -1989.3$ psi。]

第3章　数据的输入与输出

本章将介绍在MATLAB命令窗口中显示数值结果的方法，以及如何存储或从文件中查找数据的方法。

3.1　字符串和注释的输出

3.1.1　创建字符串

字符串是字符、数字和特殊字符的集合，通常用于在命令窗口中注释数据或图形。字符串的创建、存储和运算与向量或矩阵类似，不同点在于字符串的定义放在一对单引号之内，而且每个字符都占据一个元素的位置。

考虑下面的例子，令 *s* 等于字符串'testing123'。MATLAB将这个字符串定义为向量：

```
s = 'testing123'
```

或

```
s = ['testing123']
```

在向量 *s* 中，单引号内的每个字符占一个存储地址。因此，字符串的长度是10。要想获得字符串 *s* 中的特定字符，可以使用表达式，如：

```
s(7)→g
s(3:6)→stin
```

字符串的运算与数值相似。例如，程序：

```
s = 'testing123'
fs = fliplr(s)
```

执行后：

```
fs =
321gnitset
```

字符串的连接(形成更长的字符串)与数值相似。则：

```
sc = ['testing123','testing123']
```

产生(1×20)的字符串：

```
sc =
testing123testing123
```

而程序：

```
scs = ['testing123';'testing123']
```

产生(2×10)的矩阵:

```
scs =
testing123
testing123
```

因此,

```
scs(1,:)→testing123
scs(2,:)→testing123
```

注意,scs 每行具有相同数目的字符(列数)。

要找到字符串中的位置可以使用:

```
findstr(string1,string2)
```

找到两个字符串中较短的一个在较长的一个中的位置。用下面的程序可以找到'123'在另一个字符串中的位置。

```
sc = ['testing123','testing123']
Loc = findstr(sc,'123')
```

执行程序后得到:

```
sc =
testing123testing123
Loc =
  8   18
```

因此,'123'第一次出现的位置是 8,第二次出现的位置是 18。如果字符不存在,则 *Loc* 等于空向量,即[]。

如果将字符串放在矩阵中的每一行,有一个很方便的方法可得到字符串表达式。这要求每一行必须包含相同数目的字符。当字符串表达式的长度不相等时,可以使用空格填充字符串的其他部分,如:

```
lab = ['first  ';'last   ';'middle']
```

则,

```
lab(1,:)→firstb
lab(2,:)→lastbb
lab(3,:)→middle
```

其中,*b* 为空格。MATLAB 提供的函数

```
char
```

同样可以完成这一功能。因此,上面的表达式可以方便地用下面的表达式代替:

```
lab = char('first','last','middle')
ord = size(lab)
```

执行后得到：

```
lab =
first
last
middle
ord =
  3   6
```

其中，每一个字符串表达式为矩阵 *lab* 的一行，形成一个(3×6)维的数组。在命令窗口显示中看不到末尾的空格。要表示字符串末尾的空格，通常使用函数：

```
deblank
```

3.1.2 数值转换成字符串并显示

将数值转换为字符串可以使用：

```
z = num2str(num)
```

其中，*z* 是一个字符串变量，而 *num* 为数值、数值矩阵或是数值及矩阵的表达式。函数 num2str 经常用于在 MATLAB 命令窗口或图形中显示数值输出的注释。

一个典型结构可将转换的数值与一些相关的文本关联起来。因此，若 *num* 是用千克表示的重量，则在 MATLAB 命令窗口中用 disp 函数显示 *num* 如下：

```
num = 12.567;
disp(['Product weight = ' num2str(num) 'kg'])
```

在 num2str 两边至少要留有一个空格。执行后，显示在 MATLAB 命令窗口中的结果为：

```
Product weight = 12.567 kg
```

实质上，这个字符串是长度为 26 的向量。注意，空格没有被删除。

若 *num* 是重量的向量，则：

```
num = [12.567,3.458,9.111];
disp(['Product weight = ' num2str(num) 'kg'])
```

显示：

```
Product weight = 12.567   3.458   9.111 kg
```

然而，要建立与每个 *num* 值一致的注释，可以使用 repmat 函数，如：

```
num = [12.567,3.458,9.111];
n = length(num);
disp([repmat('Product weight = ',n,1)num2str(num') repmat(' kg',n,1)])
```

执行后显示：

```
Product weight = 12.567 kg
Product weight = 3.458 kg
Product weight = 9.111 kg
```

若不需要显示 *num* 的注释,则:

```
num=[12.567,3.458,9.111];
disp(num)
```

在 MATLAB 命令窗口中显示:

```
12.5670   3.4580   9.1110
```

而

```
num=[12.567, 3.458, 9.111];
disp(num')
```

显示:

```
12.5670
3.4580
9.1110
```

MATLAB 允许用户指定数值转换为字符串时的位数:

```
num2str(a,N)
```

其中,a 是要转换的数值,N 是数值的位数。若指定的位数小于小数点左边的位数,则 MATLAB 将数值转换为指数描述形式,且有效数字等于 N。

考虑下面的例子,$a=1000\pi=3141.592\,653\,589$,则:

```
num2str(a,1)→3e+003
num2str(a,3)→3.14e+003
num2str(a,4)→3142
num2str(a,5)→3141.5
num2str(a,8)→3141.5927
```

注意,位数不包括小数点。

函数 fprintf 可以在 MATLAB 命令窗口中显示有格式的数据。相对于函数 disp,其优势在于能够控制数值的显示格式。在 MATLAB 命令窗口中,函数 fprintf 的语法结构为:

```
fprintf(1,'%....',variables)
```

其中的第一个参数为'1',表示在 MATLAB 命令窗口中输出,而引号内的参数是变量格式的附加说明。当 *variables* 为向量或矩阵时,格式化说明按列循环应用。格式说明的顺序对应着变量的顺序。符号% 放在每个具体格式说明之前。常用的格式说明形式为:

```
x.yf
```

f 为格式类型中的一种。(fprintf 的其他格式参见 MATLAB 的帮助文件。)x 为给定数值整数部分的位数,而 y 是小数点后的位数。通过以下向量可以说明 fprintf 的几种不同用法。

```
num=[12, -14, 3.458, 0.11167];
```

用 fprintf 函数在一行中显示该向量:

```
num=[12, -14, 3.458, 0.11167];
```

```
fprintf(1,'%5.3f  ',num)
```

结果为:

```
12.000  -14.000  3.458  0.112
```

注意,*num*(1)和 *num*(2)的值加了 3 个 0,而 *num*(4)在小数点后则近似取了 3 位数。数值之间有两个空格,这通过在 fprintf 参数中 *f* 与撇号之间留有两个空格得到。如果要将这 4 个数值作为一列来显示,可以采用定义符"\n",如下所示:

```
num=[12 -14 3.458 0.11167];
fprintf(1,'%5.3f\n',num)
```

结果为:

```
12.000
-14.000
3.458
0.112
```

若想得到与给定精度相同的 4 个数值,必须给每一个量一个格式说明。程序为:

```
num=[12, -14, 3.458, 0.11167];
fprintf(1,'%2.0f  %2.0f  %5.3f  %5.5f',num)
```

其中,每个 *f* 和% 之间都有两个空格,这样数值就被两个空格分开了。程序执行后产生:

```
12  -14  3.458  0.11167
```

可以用以下方法对每一个数值进行注释[①]:

```
num=[12, -14, 3.458, 0.11167];
fprintf(1,'weight=%2.0f kg pressure=%2.0f kPa time=%5.3f s length=%5.5f
        m',num)
```

程序执行结果为:

```
weight=12 kg pressure= -14 kPa time=3.458 s length=0.11167 m
```

如果想按列显示数值,可将前面的程序修改为[②]:

```
num=[12, -14, 3.458, 0.11167];
fprintf(1,'weight=%2.0f kg \npressure=%2.0f kPa \ntime=%5.3f s \nlength
       =%5.5f m',num)
```

程序执行结果为:

```
weight=12 kg
pressure= -14 kPa
time=3.458 s
length=0.11167 m
```

要让每个数值按相同的格式显示,可以简化格式说明并产生注释输出,程序如下:

---

① fprintf 语句不能像显示的那样断开书写。之所以写为两行是由于纸面宽度的限制。

② 同上。

```
num=[12, -14, 3.458, 0.11167];
nn=1:length(num);
fprintf(1,'x(%1.0f)=%7.5f\n',[nn;num])
```

结果为:

```
x(1)=12.00000
x(2)=-14.00000
x(3)=3.45800
x(4)=0.11167
```

函数 num2str 在%后加入格式化声明,代替 fprintf 函数的第二个参数 N(即数据位声明)可实现格式化显示。如果将一个很小的数显示为 0,而不以指数形式显示,则当 $x=0.000\,45$ 时,程序为:

```
x=0.00045;
disp(['x='num2str(x,'%2.1f')])
```

显示:

```
x = 0.0
```

而

```
x=0.00045;
disp(['x='num2str(x,1)])
```

显示:

```
x = 0.0004
```

3.2 用 INPUT 函数输入数据

数组可以按程序或函数限定的格式由用户用 input 输入。此外,input 还可以在 MATLAB 命令窗口中显示信息以提示用户输入什么。然而,数据的具体形式取决于数据是标量、向量还是矩阵,以及是数值还是字符串。举例说明以下这几种情况。其他数据输入方法参见 3.3 节、3.5 节和 5.2 节。

3.2.1 标量的输入

输入一个数值时,可以采用:

```
InputData=input('Enter the temperature in degrees C:');
```

在 MATLAB 命令窗口中显示:

```
Enter the temperature in degrees C: 121.7
```

其中,数值 121.7 由用户输入。表达式末尾的分号可禁止系统对于输入数据的回显。变量 *InputData* 的值为 121.7。

在同一表达式中可以修改用户输入的数据。例如,将角度转换成弧度:

```
InputData=input('Enter the starting angle in degrees:')*pi/180;
```

结果显示:

```
Enter the starting angle in degrees:45
```

其中,数值45是由用户输入的。然而, *InputData* 的数值为0.7854(= 45π/180)。

又如,考虑温度从摄氏度向华氏度的变换,输入:

```
InputData = 1.8 * input('Enter the temperature in degrees C:') + 32;
```

结果显示:

```
Enter the temperature in degrees C:100
```

其中,数值100由用户输入。而 *InputData* 的数值为212。

如果希望将提示信息显示在几行中,可以在单引号内的适当位置上加入定义符"\n",即:

```
InputData = input('Enter the starting angle\nin degrees:') * pi/180;
```

则结果显示:

```
Enter the starting angle
in degrees:
```

注意,在输入的字符串中,'\n'与'in'之间没有空格。这样做能正确显示两行。

3.2.2 字符串的输入

输入一个字符串时,在input函数后面加上's'。即:

```
InputData = input('Enter file name,including its extension:','s');
```

在MATLAB命令窗口中显示:

```
Enter file name,including its extension:DataSet3.txt
```

其中,字符串 *DataSet3.txt* 由用户输入。注意,这里不需要加单引号。*InputData* 的值是字符串 *DataSet3.txt*,长度为12的向量。

3.2.3 向量的输入

输入由数值构成的向量时,可以采用:

```
InputData = input('Enter the four temperatures in degrees C:');
```

在MATLAB命令窗口显示:

```
Enter the four temperatures in degrees C:[120, 141, 169, 201]
```

其中,向量[120, 141, 169, 201]由用户输入。这里的方括号是必需的。若要输入列向量,则用户应输入[120, 141, 169, 201]'或[120;141;169;201]。

3.2.4 矩阵的输入

输入由数值构成的矩阵时,可以采用:

```
InputData = input('Enter the three temperatures in degrees C\nfor levels 1 and 2:');
```

在MATLAB命令窗口中显示:

```
Enter the three temperatures in degrees C
at levels 1 and 2:[67, 35, 91;44, 51, 103]
```

其中,数组[67, 35, 91;44, 51, 103]由用户输入。变量 *InputData* 是一个(2×3)维数组。

3.3 数据文件的输入与输出

如前所述,数据输入的一种方法是通过程序用 input 函数完成。第二种方法是在程序中定义数据,如同 2.3 节和 2.4 节中讨论的一样。如第 5 章所述,创建数据的语句也可以在函数中出现。实际上,可以定义只包含数据的函数。参见 5.2 节中的例子。

其他输入数据的方法是将数据存放在 ASCII① 文本文件之中,然后调用 load 函数。load 函数按行读取数据,每行由回车符分隔,并且每个数值之间用空格或逗号分隔。每行中数据的列数必须相同,而每列中的行数也必须相等。这些要求与矩阵的创建非常相似。这里的回车符代替了分号。在创建行向量时,输入数据可以不用回车符。而创建列向量时每一个数值后面都要跟一个回车符。

下面举例说明 load 函数的两种使用方法。假设数据按下面的形式存放在 *DataSection33.txt* 文件中:

```
11 12 13
21 22 23
31 32 33
41 42 43
```

load 函数的用途在于将不带扩展名(后缀“.txt”)的文件名变为变量名,将文件中的数据变为向量或矩阵元素。因此,程序中名为 *DataSection33* 的变量是一个(4×3)阶的数值矩阵,在程序中的使用如同一个已放置在等号左边的名为 *DataSection33* 的变量一样。

函数语句可以为:

```
load DataSection33.txt
```

或

```
load 'DataSection33.txt'
```

或

```
load('DataSection33.txt')
```

推荐使用最后一种形式。函数应用的前提是假设文件的路径已经指定。否则,可以使用 1.3.3 节描述的步骤。这三种函数形式应用于创建程序时,文件名已知且不再更改的情况。因此,如果希望对矩阵中的每一个元素取平方,可以输入:

```
load('DataSection33.txt')
y = DataSection33.^2
```

结果显示:

① ASCII 码是用于信息交换的美国标准码,表明一种不限定格式的纯文本,用于两个计算机程序之间方便地交换数据和文本。通常所说的文本文件就是指 ASCII 码文本文件。

```
y =
     121    144    169
     441    484    529
     961   1024   1089
    1681   1764   1849
```

另一方面,如果在不同文件中运算数据,而且具有不同的文件名,则必须要使用不同的方法。因此,进行数据运算时,用户首先应通过程序或函数输入文件名。如前所述,程序执行后,给定文件中的所有元素都会取平方。相应的程序为:

```
FileName1 = input('Enter file name containing data (including suffix):','s');
load(FileName1);
m = findstr(FileName1,'.');
data1 = eval(FileName1(1:m - 1));
y = data1.^2
```

如前所述,函数 findstr 用来查找单引号内字符串中第一个出现目标字符的位置,并返回其值,上例中的运算符为句号(.)。这里使用该函数是为了限制组成 *FileName1* 的字符串,但不包括句号,因此字符串的长度为 $m-1$。这样就删除了文件名后缀。然而,去除后缀的 *FileName1* 对于程序表达式来说仍然是未知的,因此还必须将其转化为数值。这项工作由 eval 函数来完成,该函数可计算参数引用中字符串的值。

程序执行后,首先要提供文件名。使用文件 *DataSection33.txt*,即在命令窗口中显示:

```
Enter file name containing data(including suffix):DataSection33.txt
```

而 *DataSection33.txt* 是用户输入的。回车后,结果显示为:

```
y =
     121    144    169
     441    484    529
     961   1024   1089
    1681   1764   1849
```

即前面所得的结果。

如果要将程序中的数值保存到文件之中,可使用:

```
save
```

假如要将文件 *DataSection33.txt* 中每个值的平方存成 ASCII 文本文件,程序为:

```
load('DataSection33.txt')
y = DataSection33.^2;
save ('SavedDataSection33.txt', 'y','-ascii')
```

结果生成一个包含下列内容的文本文件:

```
1.2100000e+002 1.4400000e+002 1.6900000e+002
4.4100000e+002 4.8400000e+002 5.2900000e+002
9.6100000e+002 1.0240000e+003 1.0890000e+003
1.6810000e+003 1.7640000e+003 1.8490000e+003
```

给定文件名时,MATLAB 将文件存放在当前的目录中。而要将文件存放在指定的目录中,则必须给出完整的路径名。例如:

```
load ('DataSection33.txt')
y = DataSection33.^2;
save ('c:\Matlab mfiles\Matlab results\SavedDataSection33.txt', 'y','-ascii')
```

注意,一定要用单引号将完整的路径名和文件名引起来,这是因为在目录名上出现了冒号和空格。另外,若想在这个文件中保存其他的值,可以在 save 语句中按以下方法附加相应的变量名。若让上面的程序还能计算 *DataSection33.txt* 中数值的平方根,则程序变为:

```
load ('DataSection33.txt')
y = DataSection33.^2;
z = sqrt(DataSection33);
save ('c:\Matlab mfiles\Matlab results\SavedDataSection331.txt','y', 'z','-ascii')
```

程序执行后,创建文件 *SavedDataSection331.txt*,内容为:

```
1.2100000e+002 1.4400000e+002 1.6900000e+002
4.4100000e+002 4.8400000e+002 5.2900000e+002
9.6100000e+002 1.0240000e+003 1.0890000e+003
1.6810000e+003 1.7640000e+003 1.8490000e+003
3.3166248e+000 3.4641016e+000 3.6055513e+000
4.5825757e+000 4.6904158e+000 4.7958315e+000
5.5677644e+000 5.6568542e+000 5.7445626e+000
6.4031242e+000 6.4807407e+000 6.5574385e+000
```

数据的前 4 行由 y 确定,后 4 行由 z 确定。

3.4 单元矩阵

单元矩阵是一类特殊的矩阵,其元素由单元组成并包含矩阵,可用于存储相异的数据。像标准的向量和矩阵一样,单元矩阵中的单元可通过矩阵索引进行访问。但单元符号与矩阵符号不同,用大括号代替了矩阵中的方括号。下面将说明如何创建并访问单元矩阵。

创建下列四种不同数据类型的矩阵:

```
A = ones(3, 2)
B = magic(3)
C = char('Pressure', 'Temperature', 'Displacement')
D = [6+7j, 15];
```

结果得到:

```
A =
    1  1
    1  1
    1  1
B =
    8  1  6
    3  5  7
    4  9  2
C =
Pressure
Temperature
```

```
Displacement
D =
  6.0000 + 7.0000i 15.0000
```

创建(2×2)阶单元矩阵,创建过程与标准矩阵创建过程类似,只是其分隔符变为了大括号。则:

```
A = ones(3, 2);
B = magic(3);
C = char('Pressure', 'Temperature', 'Displacement');
D = [6+7j, 15];
Cel = {A, B;C, D}
```

执行结果为:

```
Cel =
  [3×2 double]  [3×3 double]
  [3×12 char]  [1×2 double]
```

注意,结果中没有给出每个单元中的确切数据,只给出了各单元中数据矩阵的阶次和数据类型。要在命令窗口中显示 *Cel* 中的内容,应使用函数:

```
celldisp
```

则修改后的程序为:

```
A = ones(3, 2);
B = magic(3);
C = char('Pressure', 'Temperature', 'Displacement');
D = [6+7j, 15];
cel = {A, B;C, D};
celldisp(Cel)
```

执行结果为:

```
Cel{1, 1} =
   1   1
   1   1
   1   1
Cel{2, 1} =
Pressure
Temperature
Displacement
Cel{1, 2} =
   8  1  6
   3  5  7
   4  9  2
Cel{2, 2} =
   6.0000 + 7.0000i   15.0000
```

可以用访问标准矩阵元素的索引符访问单元矩阵的单元,只是需要将圆括号改为大括号。因此,如果要访问 *Cel* 矩阵中第一行第二列的单元,可将程序修改如下:

```
A = ones(3, 2);
```

```
B = magic(3);
C = char('Pressure', 'Temperature', 'Displacement');
D = [6+7j, 15];
Cel = {A, B;C, D};
Cell_1_2 = Cel{1, 2}
```

执行结果在命令窗口中显示为:

```
Cell_1_2 =
   8   1   6
   3   5   7
   4   9   2
```

3.5 输入微软 Excel 文件

在微软 Excel 中创建的数据文件可用以下函数

```
[X, Y] = xlsread('Filename')
```

读入 MATLAB。

其中,***X*** 为包含全部列和行数据的矩阵,***Y*** 为包含与数据对应的文本标题的单元矩阵。文件名必须包含后缀'.xls'。

ForceDispData.xls

	A	B	C	D	E
1	Transducer Linearity				
2	Force	Displacement			
3	(kPa)	(mm)			
4	100	0.1			
5	110	0.2			
6	135	0.33			
7	150	0.4			
8	175	0.55			
9					
10					
11					

图 3.1 微软 Excel 中的数据

为说明如何应用该函数,考虑在 Excel 中创建包含如图 3.1 所示数据的文件 *ForceDispData.xls*。则读取文件数据的程序为:

```
[X, Y] = xlsread('ForceDispData.xls')
```

假定在本语句之前已设定文件路径,则程序执行后,在命令窗口显示以下数据:

```
X =
  100.0000 0.1000
  110.0000 0.2000
  135.0000 0.3300
  150.0000 0.4000
  175.0000 0.5500
```

```
Y =
    [1x20 char]        []
    'Force'      'Displacement'
    '(kPa)'      '(mm)'
```

单元矩阵 Y 为(3×2)阶矩阵。因为 Excel 第一行第一列(Excel 单元 A1)中仅包含一行文本,而第二列(Excel 单元 B1)中没有文本,因此 MATLAB 没有将其作为列标题的一部分,而只是收集了其文本信息,但并没有显示。在命令窗口中输入 Y{1, 1}可以访问该文本,结果如下:

```
ans =
Transducer Linearity
```

3.6 应用函数一览

表 3.1 给出了本章应用到的函数介绍及其说明。

表 3.1　第 3 章应用函数介绍

MATLAB 函数	说明
char	将字符串放入矩阵行,必要时可加入空格
celldisp	在命令窗口显示单元矩阵内容
deblank	去掉字符串表达式末尾的空格
eval	执行包含 MATLAB 表达式的字符串
disp	在命令窗口显示文本或矩阵
findstr	在字符串中查找指定字符串
fprintf	将格式化数据写入文件或命令窗口
input	在命令窗口中要求用户输入信息
load	从文件中引入变量
num2str	将数值或数值矩阵转换为字符串
save	将特定变量保存到文件中
xlsread	从微软 Excel 中读取文件

练习

3.1　由下式产生黄金分割数:

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right] \quad n = 0,1,2,\cdots$$

使用 fprintf 和 disp 产生前 16 个数值,并在 MATLAB 命令窗口中显示以下内容:

```
F0 = 0
F1 = 1
F2 = 1
F3 = 2
⋮
F15 = 610
```

3.2　编写一段程序实现从英制单位英尺到公制单位米的转换。显示在命令窗口中的结

果应该如下所示:

```
Enter the value of length in feet:11.4
11.4 ft = 3.4747 m
```

其中,11.4是由用户输入的数值。

3.3 每英亩为43 560平方英尺。编写一段程序实现英亩到平方米的转换。显示在命令窗口中的结果应该如下所示:

```
Enter the number of acres:2.4
2.4 acres = 9712.4554 sq.m
```

其中,2.4是由用户输入的数值。

3.4 编写一段程序将小于 2^{52}(4.5036×10^{15})的整数转换为二进制数。在MATLAB中,应用函数 dec2bin 进行转换操作,函数参数为十进制数,输出值为二进制字符串。显示在命令窗口中的结果应该如下所示:

```
Enter a positive integer < 4.5x10^15:37
The binary representation of 37 is 100101
```

其中,37是由用户输入的数值。

3.5 编写一段程序显示复数模并用弧度显示其相角。显示在命令窗口中的结果应该如下所示:

```
The magnitude and phase of  -7+13i is
 Magnitude = 14.7648      Phase angle = 118.3008 degrees
```

其中,-7 + 13*i* 是由用户输入的数值。

第4章　程序流程控制

本章介绍控制程序表达式运算顺序的多种方法，以及一系列典型的用以实现条件和逻辑控制的算法。

4.1　概述——逻辑运算符

程序表达式的运算顺序有4种结构控制命令：while，if，for 和 switch。每当这些命令出现时，在程序中的稍后部分都必须跟随一个 end 语句。只有当所有条件满足时，控制结构命令与 end 命令之间的语句才会被执行。每一种控制结构命令都可以随意相互之间嵌套或自身嵌套，这称为嵌套结构。

控制结构通常将关系运算或逻辑运算的结果当做判定条件。当条件满足时，控制语句将程序转向特定部分执行相应的语句。MATLAB 的部分关系运算符和逻辑运算符参见表4.1。

表4.1　部分关系运算符及逻辑运算符

条件	数学符号	MATLAB 符号
关系运算符		
等于	=	==
不等于	≠	~=
小于	<	<
大于	>	>
小于等于	≤	<=
大于等于	≥	>=
逻辑运算符		
与	AND	&
或	OR	|
非	NOT	~

采用控制结构时，建议控制结构命令与 end 命令对齐，中间部分缩进，形成一个锯齿状结构，这将增强程序或函数的可读性。采用嵌套结构时，内部嵌套部分也采用锯齿状结构，并保证整个嵌套结构都是锯齿状的。这样的语句结构在 MATLAB 的编辑/调试窗口可以自动实现。

利用表4.1中的逻辑运算符可产生这样一个逻辑函数：逻辑表达式结果为真时输出为1，结果为假时输出为0。要产生以下的函数 $g(x)$：

$$
\begin{aligned}
g(x) &= f(x) \qquad a \leq x < b \\
&= 0 \qquad x < a \text{ 和 } b \geq x
\end{aligned}
$$

则逻辑表达式可写为：

```
y = (a <= x & x < b);
```

a 和 b 的值已知，且

```
(a<= x & x<b)
```

当 $x \geqslant a$ 和 $x<b$ 时,逻辑表达式的值为 1。当 x 为其他值时,逻辑表达式的值为 0。因此,如果令 $a=-1,b=2,f(x)=e^{x/2}$,且 $x=[-4\ \ -1\ 1\ 4]$,则采用这一逻辑表达式的程序如下:

```
a= -1;b=2;
x=[-4, -1, 1, 4];
gofx = exp(x/2).*(a<= x & x<b)
```

最后计算得出:

```
gofx =
     0  0.6065  1.6487     0
```

由于 x 为(1×4)阵列,故程序中应用了点乘计算。

逻辑运算符可用于创建单位阶跃函数,函数 $u(t)$ 定义为:

$$u(t)=1 \quad t \geqslant 0$$
$$=0 \quad t<0$$

如果变量 t 在 $-1 \leqslant t \leqslant 1$ 范围内以 0.25 的增量变化,则下列程序代码可创建单位阶跃函数:

```
t= -1:0.25:1;
UnitStep=(t>=0);
disp('   t      UnitStep')
disp([t' UnitStep'])
```

执行结果在命令窗口显示如下:

```
      t     UnitStep
  -1.0000       0
  -0.7500       0
  -0.5000       0
  -0.2500       0
        0    1.0000
   0.2500    1.0000
   0.5000    1.0000
   0.7500    1.0000
   1.0000    1.0000
```

4.2 程序流程控制

程序流程控制通过分支或循环实现。分支控制通过 if 和 switch 语句实现,而循环则通过 for 或 while 语句实现。下面讨论这四种控制结构。

4.2.1 分支——if 语句

if 语句为条件语句,通过满足不同的条件表达式将程序转向不同的部分执行。

if 语句的常见形式如下:

```
if 条件 #1
  表达式 #1
```

```
elseif 条件 #2
  表达式 #2
else
  表达式 #3
end
```

若条件 #1 满足,则执行表达式 #1,然后执行与之对应的 end 语句后面的语句。当条件 #1 不满足时,检验条件 #2 是否满足。如果此时条件 #2 满足,则执行表达式 #2,然后执行与之相应的 end 语句后面的语句。如果条件 #1 和条件 #2 都不满足,则执行表达式 #3,然后执行 end 语句后面的语句。这里,elseif 和 else 语句是可选的,且程序中可同时出现多个 elseif 语句。

下段程序解释 if 语句结构的用法。变量 j, x 和 *nnum* 已在前面赋值或已在前面的程序中由计算过程赋值。

```
if j == 1
   z = sin(x);          ←—— 仅当j=1时执行。
   if nnum<=4           ←—— 仅当j=1时才考虑这个if语句。
     nr = 1;      ┐
     nc = 1;      ┘     ←—— 这两条语句仅当j=1且nnum≤4时才被执行。
   else
     nr = 1;      ┐
     nc = 2;      ┘     ←—— 这两条语句仅当j=1且nnum>4时才被执行。
   end
else
   nr = 2;        ┐
   nc = 1;        ┘     ←—— 这两条语句仅当j≠1时才被执行。
end
```

注意,这里内嵌了一个 if 语句,因此需要书写另一个 end 语句。并且应注意如何在程序的不同嵌套结构内部缩进表达式以增强代码的可读性。

当不满足指定条件时,可使用 error 函数终止某个程序(或某个函数,参见第 5 章)的执行。error 函数通常用于确保程序获得有效的分支数值。当程序执行到 error 函数时,会在命令窗口显示函数所包含的信息,之后终止程序或函数的执行,并将程序控制权返回到命令窗口命令行。error 函数可用于程序或函数的任意位置,并不限于 if, for, switch 和 while 结构。其应用实例参见例 4.1。

例 4.1 疲劳强度计算

根据给定的校正系数计算金属的疲劳强度。

因素	范围	校正系数
受力方式	弯曲	$C_{load} = 1$
	轴向	$C_{load} = 0.70$
尺寸	$d \leq 8$ mm	$C_{size} = 1$
	$8 \leq d \leq 250$ mm	$C_{size} = 1.189d^{-0.097}$
温度	$T < 450$℃	$C_{temp} = 1$
	450℃ $\leq T$	$C_{temp} = 1 - 0.0032(T - 840)$

判别这些因素的程序如下所示。其中,变量 *lode*, d 和 *temp* 已在前面的程序中赋值或计算出结果。*lode* 为字符型变量。

```
if lode == 'bending'
  cload = 1;
elseif load == 'axial'
  cload = 0.7;
else
  error('No such loading')
end
if d<0
  error('Negative diameter not allowed')
elseif d<=8
  csize = 1;
else
  csize = 1.189 * d^( - 0.097);
end
if temp <= 450
  ctemp = 1;
else
  ctemp = 1 - 0.0032 * (T - 840);
end
```

注意,程序中加入了几条语句以确保数据有效,否则会在命令窗口中显示错误信息并终止程序执行。

4.2.2 分支——switch 语句

switch 结构可替代系列 if-elseif-else-end 结构。

通常,switch 的语法结构为:

```
switch switch _ expression
  case case _ expression #1
     statements #1
  case case _ expression #2
     statements #2
       ⋮
  case case _ expression #n
     statements #n
  otherwise
     statements #n +1
end
```

当 *case_expression #j* = *switch_expression* ($j=1,2,\cdots,n$)成立时,执行 *statements #j* 语句。只有一个 case 被执行。执行完 *statements #j* 后,将执行 end 之后的语句。如果所有 *case_expression* 都不满足,则执行 *statements #n + 1* 语句。当然,otherwise 语句不是必需的。若 otherwise 不存在且所有 *case_expression #j* 都不满足,则执行 end 后面的语句。

下面的 switch 结构阐述了这种执行方式。变量 k 已赋值或已出现过。

```
a = 3;
switch k
 case 1
  disp('Case 1')          ←—仅当 k = 1 时执行。
 case{2,3}
```

```
  disp('Case 2 or 3')         ←—仅当 k=2 或 3 时执行。
 case a^2
  disp('Case 9')              ←—仅当 k=9 时执行。
 otherwise
  disp('Otherwise')           ←—仅当 k≠1,2,3 或 9 时执行。
end
```

例 4.2　选择平面的 4 个视图之一。

根据判断条件应用函数 switch 选择平面 $z(x,y)$在 4 个方向上的视图，满足'reg'条件时为正视图，满足'top'条件时为顶视图，满足'rside'条件时为右视图，满足'lside'条件时为左视图。surf 函数用于绘制以 x 和 y 为变量的函数 z 的三维视图，view 函数用于设定角度，程序要求用户输入选定视图进行显示。变量 x,y 和 z 已被赋值或已计算得出其值。应用 switch 函数的程序段为：

```
surf (x,y,z)
str = input('Enter view:reg,top,rside,lside','s')
switch s
  case 'reg'
    view ( - 37.5,30)
  case 'top'
    view( - 90,90)
  case 'rside'
    view(0,0)
  case 'lside'
    view( - 90,0)
  otherwise
    error('No such view')
end
```

4.2.3　for 循环

for 循环可按指定次数重复执行一系列语句，其常见结构为：

```
for 变量 = 表达式
  循环语句
end
```

这里的循环语句可以是一条或多条，并且可以是变量的函数。

下面给出 for 循环的应用示例。

例 4.3　计算借款利息

已知借款总额为 L，借期 m 个月，年利率百分数为 I_a，求借款总利息。月还款额 p_{mon} 由下式得出：

$$p_{mon} = \frac{iL}{1-(1+i)^{-m}}$$

其中，

$$i = I_a/1200$$

是以十进制表示的月利率。每月偿还部分借款，偿还的钱中一部分用于支付利息，余下部

分算做偿还借款。每月还贷后的余额称做收付差额。这些关系用数学表达式表示如下。设 $b_0 = L$,则:

$$
\begin{aligned}
i_n &= ib_{n-1} \\
p_n &= p_{mon} - i_n \qquad n = 1,2,3,\cdots,m \\
b_n &= b_{n-1} - P_n
\end{aligned}
$$

其中,i_n 是每月偿还利息 p_{mon} 的一部分,p_n 是每月偿还的本金,b_n 是未偿还的本金。则总利息为:

$$i_T = \sum_{j=1}^{m} i_j$$

计算总利息 i_T 的程序为:

```
loan = input('Enter loan amount:');                      ]
durat = input('Enter term of loan in months:');          ] ←—输入
int = input('Enter annual interest rate:')/1200;         ]
ints = zeros(1,durat);                                   ]
prins = ints;                                            ]
bals = ints;                                             ] ←—初始化
pmon = (loan * int)/(1 - (1 + int)^( - durat));          ]
bals(1) = loan;                                          ]
for m = 2:durat + 1                                      ]
  ints(m) = int * bals(m - 1);                           ]
  prins(m) = pmon - ints(m);                             ] ←—计算
  bals(m) = bals(m - 1) - prins(m);                      ]
end                                                      ]
fprintf(1,'Total interest = $ %8.2f',sum(ints))          ←—输出
```

正如 1.3.3 节所述,该程序满足程序的一般结构:输入、初始化、计算和输出。本例的执行结果显示在 MATLAB 命令窗口中。程序执行结果为:

```
Enter loan amount:100000
Enter term of loan in months:360
Enter annual interest rate:8
Total interest = $164155.25
```

上面的前三行是用户对程序中连续显示的 3 个提问的应答,用户在每个提问后输入数据,最后一行是程序结果。注意,第一个输入数值 100000 的后面没有逗号,因为逗号即意味着表达式的结束。

MATLAB 建议在进入 for 循环和 for 嵌套循环体之前,利用子程序创建变量和数组。并利用 zeros 函数指定数组大小。这将大大缩短这段程序的执行时间,否则 MATLAB 将在执行 for 循环体时自动动态分配内存,从而导致数据覆盖。这段程序的初始化部分在第 4 行到第 6 行执行。

例 4.4 find 函数的实现

给定由正数和负数构成的任意长度向量 g,用程序实现下式:

```
indx = find(g > a)
```

这里,a 是由用户指定的一个值。本例中 $a = 4$,向量 $g = [4,4,7,10,-6,42,1,0]$。程序如下:

```
g=[4, 4, 7, 10, -6, 42, 1, 0];
a=4;k=0;
indx=[];
for n=1:length(g)
 if g(n)>a
  k=k+1;
  indx(k)=n;
 end
end
disp(['Element locations for g(n)>'num2str(a)':'num2str(indx)])
```

执行上述程序后,MATLAB命令窗口显示:

```
Element locations for g(n)>4:3 4 6
```

另一种生成 *indx* 的方法是使用列扩展(参见2.4节的后面部分):

```
g=[4, 4, 7, 10, -6, 42, 1, 0];
a=4;k=0;
indx=[];
for n=1:length(g)
 if g(n)>a
  indx=[indx, n];
 end
end
disp(['Element locations for g(n)>'num2str(a)':'num2str(indx)])
```

例4.5 cumsum函数的实现

给定任意长度的向量 $\boldsymbol{c}$,用程序实现:

```
Csum=cumsum(c)
```

用向量 $\boldsymbol{c}$=[4, 4, 7, 10, -6, 42, 1, 0]来检验该程序。程序如下:

```
c=[4, 4, 7, 10, -6, 42, 1, 0];
Csum(1)=c(1);
for k=2:length(c)
  Csum(k)=Csum(k-1)+c(k);
end
disp(['Cumsum of c=' num2str(Csum)])
```

执行以上程序后,MATLAB窗口显示:

```
Cumsum of c=4 8 15 25 19 61 62 62
```

另一种生成 *Csum* 的方法为使用列扩展,程序如下:

```
c=[4, 4, 7, 10, -6, 42, 1, 0];
Csum(1)=c(1);
for k=2:length(c)
 Csum=[Csum, Csum(k-1)+c(k)];
end
disp(['Cumsum of c=' num2str(Csum)])
```

例4.6 diag函数的实现

给定($n \times n$)阶矩阵 $\boldsymbol{b}$,程序第一步实现:

```
v=diag(B)
```

第二步实现:

```
D=diag(v)
```

这里,v 是第一步得到的结果。用矩阵 $\boldsymbol{B}$=magic(4)检验程序,并假设不使用冒号和 zeros 函数。程序如下:

```
B=magic(4);
[r, c]=size(B);
for k=1:r
 v(k)=B(k,k);
end
disp(['Diagonal elements of B='])
disp(num2str(v))
for n=1:r
 for m=1:r
  if n==m
   D(n,m)=v(n);
  else
   D(n,m)=0.0;
  end
 end
end
disp(['Diagonal matrix D='])
disp(num2str(D))
```

执行完上述程序后,MATLAB 命令窗口显示:

```
Diagonal elements of B=
16
11
 6
 1
Diagonal matrix D=
16  0  0  0
0   11 0  0
0   0  6  0
0   0  0  1
```

如果允许使用冒号和 zeros 函数,则上述程序可简写为:

```
B=magic(4);
[r, c]=size(B);
for k=1:r
 v(k)=B(k,k);
end
disp(['Diagonal elements of B='])
disp(num2str(v'))
D=zeros(r,c);
for n=1:r
 D(n,n)=v(n);
end
disp(['Diagonal matrix D='])
disp(num2str(D))
```

例 4.7 指定阵列元素

创建($n \times n$)阶矩阵,其所有元素值都为 +1 或-1,且每一个元素值符号都与四周相邻的元

素值符号不同。n 为任意值。矩阵形为：

$$M=\begin{bmatrix} 1 & -1 & 1 & \cdots \\ -1 & 1 & -1 & \cdots \\ 1 & -1 & 1 & \cdots \\ \vdots & & & \end{bmatrix} \rightarrow (n \times n)$$

程序为：

```
n = input('Enter the order of the square matrix:');
k = 1:n;
M = zerors(n,n);
OddRow = (-1).^(k-1);
EvenRow = (-1).^k;
for m = 1:2:n
  M(m,:) = OddRow;
  if m+1 <= n
    M(m+1,:) = EvenRow;
  end
end
M
```

$n = 3$ 时程序在命令窗口中的显示结果为：

```
Enter the order of the square matrix:3
M =
   1  -1   1
  -1   1  -1
   1  -1   1
```

其中，3 为用户输入值。

例 4.8　用区域对分法求多个根①

MATLAB 的 fzero 函数可求取使 $f(x)$ 接近 0 的 x 值，即 fzero 可用于估计根的区域（参见5.5.1节）。但是，该函数一次只能求得一个根，要求得其他的根必须重复调用该函数。假设要自动求得使 $f(x)=0$ 的一系列 $x(x_1,x_2,\cdots)$ 的正值。当然，$f(x)$ 有多个根，且随 x 的增加 $f(x)$ 的值也将改变。

自变量区域对分法是一种方程求根方法②。该方法的求根原理参见图 4.1。给独立变量 x 赋初值 $x=x_{start}$，可以从图中直接看出 $f(x_{start})$ 的符号。然后，变量 x 加增量 Δ，$f(x_{start}+\Delta)$ 的符号也可以确定。比较这两个值的符号，若符号相同（图 4.1 中即如此），x 再加 Δ，判定 $f(x_{start}+2\Delta)$ 的符号并将其与 $f(x_{start})$ 的符号比较。若符号不相同，则 x 的值减少间隔的一半，即 $\Delta/2$。从图 4.1 中可以看出，当 $x=x_{start}+3\Delta$ 时，$f(x)$ 的符号改变了，则下一个 x 为 $x=x_{start}+5\Delta/2$。再将 $f(x_{start}+5\Delta/2)$ 的符号与 $f(x_{start})$ 的符号比较。如果两个符号相同，则 x 增加当前间隔的一半，否则 x 减少间隔的一半。在本例中，$f(x_{start})$ 与 $f(x_{start}+5\Delta/2)$ 符号相同，则下一个 x 值为 $x_{start}+11\Delta/4$。重复以上过程，直到当前的增加量与当前的 x

---

① 参见 S. C. Chapra 和 R. P. Canale, *Numerical Methods for Engineers*，第二版，McGraw-Hill, New York, 1988 年，第 128ff 页。

② 尽管这种方法可以求出根，但并非是最佳的方法。对于在 fzero 中所使用的方法，要想在给定精度上找到根，需要 2～3 次的作用。

的比值小于某一限值,即 $\Delta_{current}/x_{current} < t_o$。当满足这一限值标准时,根 $x_1 = x_{current}$。重复求根过程,直到满足所要求的根的个数为止。求得一个根 x_j后,设 $x_{start} = 1.05x_j$,将 Δ 设为最初值,重复求根过程。

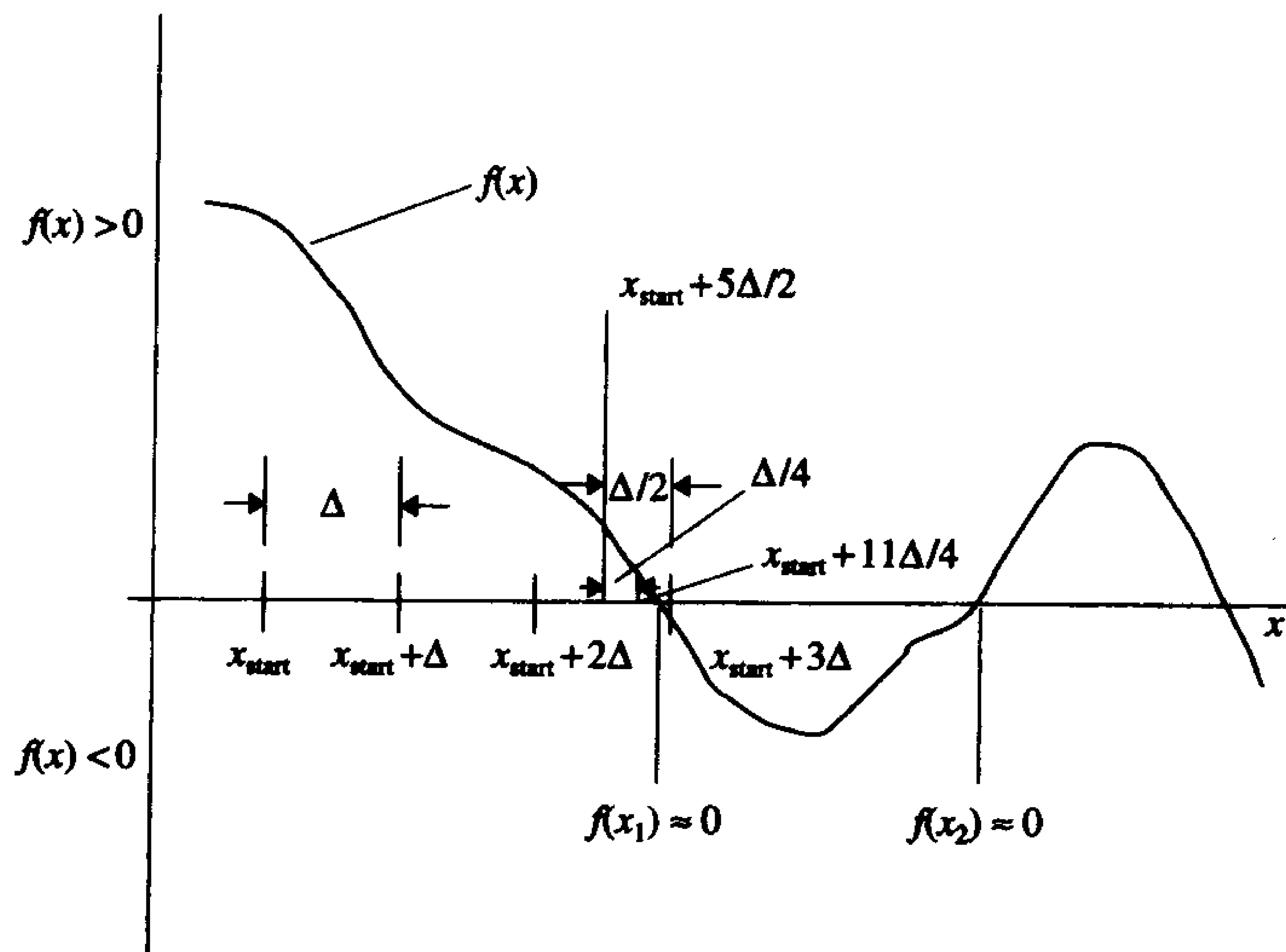


图 4.1　区域对分法

写一段程序,采用区域对分法求满足下式的前 5 个根:

$$f(x) = \cos(ax) \cong 0$$

这里,a 是一个常数。当 x_j增加的量与 x 的比值小于 $t_o = 10^{-6}$时,x_j 是所求的根,并满足 $x_j \geqslant x_{start} \geqslant 0$,且 $j = 1,2,3,\cdots,n$。因此,求根程序的输入参数为 n, x_{start}, t_o和 Δ,对特定的 $f(x)$还有常数 a。

本例中,$n=5, t_o=10^{-6}, x_{start}=0.2, \Delta=0.3, a=\pi$。程序如下:

```
n = 5;a = pi;
increment = 0.3;tolerance = 1e - 6;
xstart = 0.2;
x = xstart;
dx = increment;
for m = 1:n
 s1 = sign(cos(a * x));
 while dx/x > tolerance
  if s1 ~= sign(cos(a * (x + dx)))
   dx = dx/2;
  else
   x = x + dx;
  end
 end
 route(m) = x;
 dx = increment;
 x = 1.05 * x;
end
disp(route)
```

函数 sign 当参数符号为正、负或零时分别返回 +1, -1 和 0。程序执行结果在命令窗口显示如下：

```
0.5000    1.5000    2.5000    3.5000    4.5000
```

在 5.3 节中，将这段程序转换成一个函数，以便求解 $f(x)$的根。

4.2.4 while 循环

while 循环无限次执行一条或多条语句，当条件满足时停止循环。其语法结构为：

```
while 条件
  语句
end
```

其中，定义“条件”的表达式中可包含由“语句”计算得出的一个或多个变量。

现在，给出两个 while 循环的例子。第一个例子用来确保用户在规定的区间内输入数据。第二个例子用于判定是否收敛。

例 4.9　正确输入数据

下列摘录的程序段要求用户连续输入 1 到 8 之间的数，若所输入的数值超出范围，则程序终止。其中，input 函数将引号内的部分显示于 MATLAB 窗口，并等待用户输入数据，再将所输入的数据赋给变量 *nfnum*。符号“|”代表逻辑或。

```
nfnum = 0;
while(nfnum < 1)|(nfnum > 8)
  nfnum = input('Enter a number from 1 to 8: ');
end
```

对变量 *nfnum* 赋初值(本例中为 0)，while 命令检查条件表达式(*nfnum* < 1)|(*nfnum* > 8)是否满足，并以此决定是否进入 while 循环。执行完 end 语句前的最后一行语句后，程序返回，重新判断 while 条件表达式是否满足。若条件不满足，重复执行 while 后面的语句，否则执行 end 后面的语句。注意，在进入 while 循环块前一定要给变量 *nfnum* 赋初值。

例 4.10　级数收敛性判定

计算并显示级数 S_n 的项数 N, S_n 如下：

$$S_N = \sum_{n=1}^{N} \frac{1}{n^2}$$

当收敛到终值 $S_\infty = \pi^2/6$ 的 0.01%以内时，项数 N 为多少？程序如下：

```
series = 1; k = 2; exact = pi^2/6;
while abs((series - exact)/exact) > = 1e - 4
  series = series + 1/k^2;
  k = k + 1;
end
disp(['Number of terms = 'num2str(k - 1)])
```

执行结果在命令窗口显示为：

```
Number of terms = 6079
```

给变量 *series* 赋初值(本例中为 1),while 语句检查条件 abs ((*series-exact*)/*exact*)是否满足,决定是否执行 while 循环。执行完 end 的前一条语句后,程序返回到 while 开始处检查 while 条件是否满足。如果条件不满足,接着执行 while 下面的语句;否则执行 end 下面的语句。本例在 MATLAB 命令窗口中显示结果。

当出现(*series-exact*)为负值的情况时,判断仍然有效。使用((*series-exact*)/*exact*)的绝对值检查条件,负值小于 10^{-4} 是显然成立的,而其绝对值却不一定小于 10^{-4}。采用绝对值条件后不用再考虑 *series* 值是从正向还是从负向趋近极限值。在构建 while 循环的条件表达式时必须要小心,因为一旦表达式不严格,循环将会无限次地执行下去,直至按下 Ctrl^C 组合键才能终止。

4.2.5 for 循环或 while 循环的提前终止

break 函数可实现 for 循环或 while 循环的提前终止。若 break 函数嵌套于 for 循环或 while 循环的内部,则返回到上一级的 for 循环或 while 循环。考虑下列程序段:

```
for j = 1:14
  ⋮
  b = 1
  while b<25
   ⋮
   if n<0
    break        当n<0时,while循环将退出并
   end           从这个end语句之后继续执行。
   ⋮
  end   <------
  ⋮
end
```

4.3 应用函数一览

在本章开始处的表 4.1 中,介绍了几种逻辑运算符和关系运算符。最后,在表 4.2 中介绍本章应用到的其他函数。

表 4.2　第 4 章用到的函数

MATLAB 函数	说明
break	终止 for 循环或 while 循环
case	switch 结构的替代标识符
else	根据关系表达式或逻辑表达式执行语句
elseif	根据关系表达式或逻辑表达式执行语句
end	结束 for,while,if,或 switch 结构
error	显示错误信息并终止程序执行
for	按指定次数重复执行语句
if	根据关系表达式或逻辑表达式执行语句
sign	符号函数
switch	依据条件值执行不同语句
while	无限次执行语句

练习

4.1 编写程序完成 4.1 节中提到的逻辑运算，即给定任意长度的向量 $\boldsymbol{h}$，判定其元素值大于 a 和小于 b 时结果是否为 1，并将结果用向量 $\boldsymbol{v}$ 表示出来。设 $\boldsymbol{h}$ = [1 3 6 -7 -45 12 17 9]，a = 3，b = 13，检验所设计的程序。

[答案：$\boldsymbol{v}$ = [0 0 1 0 0 1 0 1]。]

4.2 估计由下式描述的 n 个采样值 x_j 的方差：

$$s_n^2 = \frac{1}{n-1}\left[\sum_{j=1}^{n} x_j^2 - n\bar{x}_n^2\right] \qquad n > 1$$

其中，

$$\bar{x}_n = \frac{1}{n}\sum_{j=1}^{n} x_j$$

这是平均值。方差由 var 决定。

编写计算 $n > 1$，x = [45 38 47 41 35 43] 时样本方差 s_n^2 的程序。

[答案：[24.5000，22.3333，16.2500，24.2000，19.9000]。]

4.3 给定 $a > 0$，假设任意初值 $x_0 > 0$，下面的关系式将限定数 $\sqrt{a}$ 在容限为 t_0 以内：

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \qquad n = 0,1,2,\cdots$$

其中，$x_{n+1} \cong \sqrt{a}$（当 $x_0 < 0$ 时，有负方根）。编写程序，当 $a = 7$ 时，在 $|x_n - x_{n+1}| < 10^{-6}$ 条件下确定 $\sqrt{a}$ 在 (a) $x_0 = 3$ 和 (b) $x_0 = 100$ 时，计算将重复多少次。第一次重复是确定 x_1。[提示：上述关系式不是 n 的显式函数。这里，下标 n 标明下一值 x_{n+1} 是当前值 x_n 的函数。因此，每次循环后新旧值都被保存，n 记录循环次数。]

[答案：(a) $n_{\text{iterations}} = 4$，(b) $n_{\text{iterations}} = 10$。]

4.4 考虑下式：

$$x_{n+1} = x_n^2 + 0.25 \qquad n = 0,1,2,\cdots,N$$

对 $x_0 = 0$，编写两段程序求 $n = 0,5,10,\cdots,200$ 时 x_n 的值。第 1 段程序采用 for 循环结构，第 2 段程序采用 while 循环结构。x_N 收敛为何值？plot 函数的第 3 个参数设为 plot(…, …, 'ks')，画出 x_n 的平方曲线图。x 轴代表 n，y 轴为 x_n。注意，对 $n = 0,1,2,\cdots,200$ 的所有 x_n 值都要计算出来，但画图时只须每隔 5 点画出即可。另外，与习题 4.3 不同的是，x_n 必须以向量形式保存，以便随时显示出其特定元素。

4.5 χ^2 统计用于拟合优度的检验（参见习题 14.5），定义为：

$$\chi^2 = \sum_{i=1}^{k} \frac{(\boldsymbol{x}_i - \boldsymbol{e}_i)^2}{\boldsymbol{e}_i}$$

其中，$\boldsymbol{e}_i$ 和 $\boldsymbol{x}_i$ 是长度为 k 的独立向量。

若 $e_i < 5$，将 e_i，$\boldsymbol{x}_i$ 分别与 $\boldsymbol{e}_{i+1}$，$\boldsymbol{x}_{i+1}$ 相加。如果 $\boldsymbol{e}_i + \boldsymbol{e}_{i+1}$ 的和仍然小于 5，再将 $\boldsymbol{e}_{i+2}$ 与 $\boldsymbol{e}_i + \boldsymbol{e}_{i+1}$ 相加。这一过程一直重复到其和大于等于 5。如 $\boldsymbol{e}_i \geq 5$ 且剩余值 $\boldsymbol{e}_{i+1}$，$\boldsymbol{e}_{i+2}$，…，$\boldsymbol{e}_k$ 的和小于 5，则将这些剩余值与 $\boldsymbol{e}_j$ 相加。

编写程序在下列条件下计算 X^2 的值。并用下列向量检验计算结果,这些向量代表了 3 种不同的情况:

(a) $x = [1,7,8,6,5,7,3,5,4]$ 且 $e = [2,6,10,4,3,6,1,2,3]$

(b) $x = [7,11,13,6]$ 且 $e = [6,10,15,7]$

(c) $x = [3,14,20,25,14,6,2,0,1,0]$ 且 $e = [4,12,19,19,14,8,4,2,1,1]$

提示:通过检验 cumsum(e)的元素,可获得最为简洁的程序。随着计算过程的逐步深入,e 的长度将有所变化。

[答案:

(a) $e_{\text{modified}} = [8,10,7,6,6]$, $x_{\text{modified}} = [8,8,11,7,12]$, $\mathcal{X}^2 = 8.8524$

(b) $e_{\text{modified}} = [6,10,15,7]$, $x_{\text{modified}} = [7,11,13,6]$, $\mathcal{X}^2 = 0.6762$

(c) $e_{\text{modified}} = [16,19,19,14,8,8]$, $x_{\text{modified}} = [17,20,25,14,6,3]$, $\mathcal{X}^2 = 5.6349$。]

4.6 给定两个多项式:

$$y(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$

和

$$z(x) = s_1 x^m + s_2 x^{m-1} + \cdots + s_m x + s_{m+1}$$

编写程序求和,即分别在 $m = n, m < n$ 和 $m > n$ 的情况下求 $h(x) = y(x) + z(x)$。多项式的求和是相同指数的系数求和。假设程序输入为向量 $p = [p_1\ p_2 \ldots p_n\ p_{n+1}]$ 和 $s = [s_1\ s_2 \ldots s_m\ s_{m+1}]$。

用下列数组检查所编写的程序:

a. $p = [1,2,3,4]$ 且 $s = [10,20,30,40]$

b. $p = [11,12,13,14]$ 且 $s = [101,102]$

c. $p = [43,54,55]$ 且 $s = [77,66,88,44,33]$

[答案:a. $h = [11,22,33,44]$;b. $h = [11,12,114,116]$;c. $h = [77,66,131,98,88]$。]

4.7 编写程序计算 2004 年、2005 年和 2006 年中的任意一天是星期几,输入格式为:month/day/year:xx/xx/xxxx。已知下列信息:2004 年 1 月 1 日为星期四,闰年;2005 年 1 月 1 日为星期六;2006 年 1 月 1 日为星期天。编写程序使其输入/输出结果以下列方式显示:

```
Enter month, day, and year in the form xx/xx/xxxx for 2004,2005,or 2006:11/20/2005
The date 11/20/2005 is the 324 day of the year and falls on a Sunday.
```

其中,11/20/2005 为用户输入信息。建议采用函数 str2num, deblank 和 findstr。

第5章 函　　数

本章论述了 MATLAB 函数的创建及其不同用法,并介绍了几个常用于求解工程问题数值解的内置函数。

5.1 概述

程序文件是一种 M 文件,而函数文件是另一类 M 文件。在 MATLAB 中,函数文件是能够创建局部独立空间的程序文件。在函数内定义的所有变量名对函数而言都是局部的,既不影响其他程序文件和函数文件中同样的变量名,又不被其所影响。所有的 MATLAB 内置函数都是这种类型。而在用户自定义函数中被指明为全局变量的变量与此不同,这将在 5.2.2 节中加以讨论。一个函数文件的第一个非注释行必须按照特定的格式书写,这将在 5.2.2 节中给出说明。典型情况是,用户创建的 MATLAB 程序由一个程序文件组成,并且调用了任意多个用户自定义函数和 MATLAB 内置函数。

5.1.1 为什么使用函数

MATLAB 的几个内置函数需要创建一些函数,除此之外,创建函数还有以下几个原因(参见 5.5 节)。其中包括:

1. 避免代码复制。
2. 限制因改变特定代码段而带来的影响。
3. 提高代码重用性。
4. 减少整体代码的复杂性,增加可读性和可管理性。
5. 隔离复杂操作。
6. 提高移植性。
7. 更易于调试和隔离错误。
8. 因为每一段程序都被“最优化”,所以可提高整体性能。

通过函数对程序的划分,可以减少对多余数据的使用。因为对每个函数而言,提供的数据必须是已知的。

5.1.2 函数的命名

函数的名字应该与函数的功能有关系。函数名的典型长度在 9 至 20 个字符之间,并且应该使用标准的或一致的规则。例如,所有的程序文件名以 *scr* 开始,但所有的函数名没有这一前缀。函数名的正确选择可有效减少函数内的注释行。可以参考 1.3.2 节建议的命名规则。

5.1.3 函数的长度

函数的代码长度从两行到几百行不等。然而,函数的长度应该由函数内部功能的粘合性所支配——也就是说,在某种程度上做某一件事情,而不是做其他任何事情。例如,sin(x)是 100%

粘合的。但是对于一个同时包括求正弦和平方根计算的函数而言,其粘合性较低,因为求正弦和求平方根这两部分相互独立,互不相关。可利用多个高粘合性函数创建一个粘合性的函数,这样做的好处是具有可靠性——低差错率。当函数的粘合程度较低时,要隔离错误很困难。

5.1.4 函数的调试

在函数(和程序)的创建过程中,应该独立检验每行代码,以确保写完几行后即可正确工作。MATLAB 特别适合于这一处理过程,并且简单地忽略每一行的分号。进一步而言,除了那些使用大规模向量和矩阵或迭代求解的程序外,忽略分号可节省大量时间。应该用一些独立的计算或估计方法进行检验。在检验/调试阶段,应当将提供中间输出的代码行注释掉,而不是删除,直到整个函数经检验能正常工作时再删除。函数经检验正确后,如果必要,应尽量减少其执行时间。然而,创建可正确执行的程序是基本目标。

5.2 创建函数

5.2.1 概述

MATLAB 函数的创建有几种方法,最常用的是由 function 命令创建的函数文件,可以通过程序、函数或命令窗口来访问。在 5.2.2 节给出了函数文件的创建方法。第二种创建函数的方法是使用 inline 命令,创建的函数局限于一个表达式,而且只能由程序或其创建函数来调用。这部分内容将在 5.2.3 节讨论。函数的第三种形式是子函数。在一个函数文件中,如果 function 关键字出现多次,则在第一个 function 出现之后定义的所有函数都称为子函数。子函数只能由函数文件的第一个函数和文件内部的其他子函数调用。子函数与 inline 所创建函数的不同之处在于,子函数可由多个表达式组成,可以减少函数文件数量的激增。有关子函数的内容参见 5.2.4 节。

5.2.2 函数文件

一个函数至少有两行代码,第一行是 MATLAB 所要求的格式。在函数程序中没有类似 end 语句这样的结束字符或表达式,而在 for,while,if 和 switch 结构中却有 end 语句。而且,M 文件的名字应该和函数名字相同,扩展名为 *.m*。

输入和输出函数的变量数目及其类型(标量、向量、矩阵字符串或细胞类型)由函数接口控制,函数接口是函数程序的第一个非注释行。一般来说,函数文件由接口行、注释、一个或多个表达式组成。接口的一般形式如下:

```
function [OutputVariables] = FunctionName(InputVariables)
% Comments
Expression(s)
```

其中,*OutputVariables* 是输出变量,变量之间用逗号隔开;*InputVariables* 是输入变量,变量之间用逗号隔开;*FunctionName* 是函数及函数文件的名称。这些名称必须以大写字母开头。有关命名的一些限制参考 1.3.2 节。该句中的 function 是保留字。函数文件可以存储在任何已定义或即将定义的路径下,文件名为 *FunctionName.m*。在会话期间访问函数时,系统必须知道

函数和程序各自的路径。可以将函数和程序放在当前目录下，或是将函数存放在使用过程保存过的一条路径下，可参考图 1.8。

MATLAB 用紧跟函数接口语句之后的注释创建此函数的帮助信息——当在 MATLAB 命令行中输入

```
help FunctionName
```

时，所有相邻的注释行都将出现在 MATLAB 命令窗口中。任何出现在 function 语句之前的注释将不作为帮助信息的部分，遇到空行或可执行表达式时将终止帮助信息。

有三种特殊情况：

1. 如果创建的函数仅仅用于绘图、在命令窗口中显示标注数据或将数据写到文件中，函数没有返回数据到调用程序（调用程序指在一个或多个表达式中使用了该函数的程序或另一函数）。在这种情况下，函数的接口行可简写如下：

```
function FunctionName(InputVariables)
```

2. 当函数仅仅用来以指定的方式存储数据时，函数不需要任何输入参数。在这种情况下，函数接口行的形式为：

```
function OutputVariables = FunctionName
```

例 5.1 中演示了这种形式。

3. 当以普通方式创建的 M 文件数量很多时，可以将程序转换为一个主函数以减少 M 文件的数量。在这种情况下，函数接口行的形式为：

```
function FunctionName
```

这种形式在本章 5.2.4 节的末尾处说明。

正确创建函数必须明白几个概念。第一，函数中使用的变量名不必和从 MATLAB 命令窗口、程序文件或另一个函数中调用此函数时使用的名字相一致。但是，圆括号内输入变量的位置确定了信息的传递顺序——调用语句中的第一个参数将值传递到函数接口定义中的第一个参数，依次类推。

第二，每个参数名对函数而言都是局部的，并且仅在函数范围内有意义。同样的名字可以在调用这一函数的程序文件中使用，也可以在此函数调用的另一个函数中使用。然而，函数语句和调用语句中的变量名必须具有相同类型，可以是标量、向量、矩阵、单元型或字符串的一种，这样可以使函数的表达式按预先设定的类型工作。例如，如果变量没有正确的阶数，两个行向量的乘法会导致一错误信息。进而，函数语句的输入变量名相当于它们出现在等号的左边。如果一个输入变量为 a，则 a 等同于表达式 $a = [\cdots]$。除非用 global 定义，否则该变量为局部变量（global 的使用以后会谈到）。下面首先说明函数的构建过程，然后列出其他几种情况。以下是函数中计算的等式。

$$x = \cos(at) + b$$

$$y = |x| + c$$

函数返回 x 和 y 的值。现在，创建一个 *ComputeXY* 函数计算这些量，文件名为 *ComputeXY.m*[①]：

---

① 本例中的注释表明了其用途。在本书给出的大量程序和函数中，为了提高程序的可读性，注释行均被省略。大多数情况下，程序中重要的部分在程序和函数的相关内容加以讨论，也可以从上下文中看到。

```
function [x,y] = ComputeXY(t,a,b,c)
% Computation of-
%   x = cos(at) + b
%   y = |x| + c
% Scalars:a,b,c
% Vector:t,x,y
x = cos(a * t) + b;
y = abs(x) + c;
```

在 MATLAB 命令窗口中输入:

```
help ComputeXY
```

则显示以下内容:

```
Computation of -
  x = cos(at) + b
  y = |x| + c
Scalars:a,b,c
Vector:t,x,y
```

表 5.1 中列出函数接口的其他几种情况,以及调用的方式。由表中的例子可以看出,必须根据函数内的使用方法确保输入和输出变量的数目和类型正确。为了响应 help 请求,函数的注释中应指明这些限制。这里,假定 ***t*** 是一向量,***a***,***b***,***c*** 是标量。

表 5.1 函数语句[u,v] = computeXY(t,a,b,c)的几种情况

函数	调用	注释

function z = ComputeXY(t,w)
x = cos(w(1) * t) + w(2);
z = [x;abs(x) + w(3)];
	t = 0:pi/4:pi;
	w = [1.4,2,0.75];
	q = **ComputeXY**(t,w);
		$w(1) = a = 1.4; w(2) = b = 2;$
		$w(3) = c = 0.75; q \rightarrow (2 \times 5)$
		$x(:) = q(1,1:5); y(:) = q(2,1:5)$

function z = ComputeXY(t,w)
x = cos(w(1) * t) + w(2);
z = [x abs(x) + w(3)];
	t = 0:pi/4:pi;
	w = [1.4,2,0.75];
	q = **ComputeXY**(t,w);
		$w(1) = a = 1.4; w(2) = b = 2;$
		$w(3) = c = 0.75; q \rightarrow (1 \times 10)$
		$x(:) = q(1:5); y(:) = q(6:10)$

function[x,y] = ComputeXY(t,w)
x = cos(w(1) * t) + w(2);
y = abs(x) + w(3);
	t = 0:pi/4:pi;
	w = [1.4,2,0.75];
	q = **ComputeXY**(t,w);
		$w(1) = a = 1.4; w(2) = b = 2;$
		$w(3) = c = 0.75; q \rightarrow (1 \times 5)$
		$x = q$; y 不可得到§

§ 许多 MATLAB 函数使用这种形式,在本书中可以看到。

在命令窗口中输入:

```
[u,v] = ComputeXY(0:pi/4:pi,1.4,2,0.75);
```

在圆括号中的位置区分参数,对此函数而言,t = [0,pi/4,pi/2,3* pi/4,pi],$a = 1.4$,$b = 2.0$,$c = 0.75$。执行这一语句,得到以下结果:

```
u =
  3.0000  2.4540  1.4122  1.0123  1.6910
v =
  3.7500  3.2040  2.1622  1.7623  2.4410
```

一般情况下,只有执行完最后一条语句时,函数才返回结果。如果想提前返回,则用以下语句实现:

```
return
```

将函数 **ComputeXY** 修改为当变量 t 长度大于等于 2 且调用语句中的参数个数为 4 时，函数才执行。这样确保用户输入的变量个数正确且 t 不是标量。用 nargin 求出输入变量的个数。如果输入有非法数据或不合适数据时，函数返回 NaN，意为“不是一个数(not a number)”。修改后的函数为(为清晰起见，去掉了注释)：

```
function [x,y] = ComputeXY(t,a,b,c)
if length(x) == 1|nargin ~= 4
  x = NaN;
  y = NaN;
  return
end
x = cos(a * t) + b;
y = abs(x) + c;
```

有时，传递给函数的变量数目很大。这种情况下，好的方法是共享程序或函数的全局内存，或是创建不同的函数访问全局变量。全局变量的定义用：

```
global
```

MATLAB 建议，书写全局变量时全部采用大写字母以与局部变量相区别。

为说明全局变量的使用，在 **ComputeXY** 函数中定义全局变量 a，b 和 c，修改后的程序如下：

```
function[x,y] = ComputeXY(t)
global A B C
x = cos(A * t) + B;
y = abs(x) + C;
```

定义语句中全局变量之间用空格而不是逗号隔开，接口行不包括 a，b 和 c 变量。调用 **ComputeXY** 的程序如下：

```
global A B C
A = 1.4; B = 2; C = 0.75;
[u,v] = ComputeXY(0:pi/4:pi)
```

函数和其调用程序中使用的全局变量名称必须相同，值也应相同。在命令窗口中显示执行结果如下：

```
u =
  3.0000  2.4540  1.4122  1.0123  1.6910
v =
  3.7500  3.2040  2.1622  1.7623  2.4410
```

此结果与前面得到的结果一致。

函数执行时，函数的参数分别在各自的位置保留其数值。可以在调用语句中插入任何结构正确的表达式。例如，用 **ComputeXY** 计算 x，y 时，假设：

$$a = \sqrt{\frac{1.8}{(1+k)^3}} \quad k = 1, \cdots, n$$

t 从 0 变到 π，每次增加 $\pi/4$；$c = 1/0.85$；b 有 n 个值，范围由 1 变到 1.4，$n = 3$。**ComputeXY** 函数如下：

```
function[x,y] = ComputeXY(t,a,b,c)
x = cos(a * t) + b;
y = abs(x) + c;
```

调用程序为：

```
n = 3;
c = linspace(1,1.4,n);
for k = 1:n
  [u,v] = ComputeXY(0:pi/4:pi,c(k),sqrt(1.8/(1 + k)^3),1/.85)
end
```

执行程序后在命令窗口中显示：

```
u =
  1.4743    1.1814    0.4743   -0.2328   -0.5257
v =
  2.6508    2.3579    1.6508    1.4092    1.7021
u =
  1.2582    0.8460   -0.0508   -0.6929   -0.5508
v =
  2.4347    2.0225    1.2273    1.8693    1.7273
u =
  1.1677    0.6217   -0.4201   -0.8200   -0.1413
v =
  2.3442    1.7982    1.5966    1.9965    1.3178
```

如果定义全局变量 a，b 和 c，再重复上述计算，修改后的 **ComputeXY** 函数如下：

```
function [x,y] = ComputeXY(t)
global A B C
x = cos(A * t) + B;
y = abs(x) + C;
```

调用程序为：

```
global A B C
n = 3;C = 1/.85;
c = linspace(1,1.4,n);
for k = 1:n
  A = c(k);
  B = sqrt(1.8/(1 + k)^3);
  [u,v] = ComputeXY(0:pi/4:pi)
end
```

执行结果与前面得到的结果一致。

最后，说明函数返回值为向量且在程序文件中重新定义为矩阵的情况。为了简化，只考虑 x 的值。函数值在一个 for 循环内被返回。程序段如下：

```
n = 4;
c = linspace(1,1.4,n);
t = 0:pi/4:pi;
p = zeros(n,length(t));
for k = 1:4
  p(k,:) = ComputeXY(t,c(k),sqrt(1.8/(1 + k)^3),1/.85);
```

```
        ⋮
    end
        ⋮
```

在上述代码中,由于 $k=1,2,3,4$,向量 t 的长度为5,所以 $p(=u=\cos(at)+b)$是一个(4×5)阶矩阵。符号 $p(k,:)$指矩阵 p 的第 k 行,其列元素值为 **ComputeXY** 返回的行向量对应列元素值。应将 p 初始化为 zeros 向量,否则易出错。

5.2.3 inline

在 MATLAB 命令窗口、程序或函数中创建局部函数时,可以使用 inline。优点是不必将其存储为一个单独文件。在运用中有几点限制:不能调用另一个 inline 函数,只能由一个 MATLAB 表达式组成,并且只能返回一个变量——显然不允许[u,v]这种形式。因而,任何要求逻辑运算或乘法运算以求得最终结果的场合,都不能应用 inline。除了这些限制,在许多情况下使用该函数非常方便。

inline 函数的一般形式为:

```
FunctionName = inline('expression','p1','p2',…)
```

其中,*expression* 本身是任何有效的 MATLAB 表达式,$p1,p2,\cdots$是出现在 *expression* 中所有变量的名字。下面的例子用于说明 inline。创建一函数 **FofX**,求解下式:

$$f(x)=x^2\cos(ax)-b$$

其中,a 和 b 是标量,x 是向量。有:

```
FofX = inline('x.^2. * cos(a * x) - b','x','a','b')
```

显示在 MATLAB 命令窗口中,有:

```
FofX =
   Inline function;
   FofX(x,a,b) = x.^2. * cos(a * x) - b
```

因为 x 是向量,所以需要用点乘。如果用一分号结束表达式,则显示将会被禁止。

在 MATLAB 命令窗口中输入以下表达式时:

```
g = FofX([pi/3 pi/3.5],4.1)
```

系统输出为:

```
g =
   -1.5483   -1.7259
```

对函数而言,inline 的形式用于很多 MATLAB 命令中。该函数要求先创建函数,然后再执行命令。几个应用实例在5.5节给出。

5.2.4 子函数

如果在一个函数文件中出现多个 function 关键字,则在第一个 function 之后定义的所有函数都称为子函数。含有第一个 function 关键字的表达式叫做主函数。主函数是由命令窗口、程序和其他函数调用的惟一函数。子函数仅能由主函数和主函数文件中的其他子函数调用。

下面,以数值型向量的平均值和标准偏差的计算为例,说明函数和子函数的使用。这里的计算方法效率相对较低,主要用来说明子函数的属性及其使用。平均值 m 和标准偏差 s 由下式给出:

$$m = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$s = \left[\frac{1}{n-1}\left(\sum_{k=1}^{n} x_k^2 - nm^2\right)\right]^{1/2}$$

主函数为 **MeanStdDev**,计算 **m** 的子函数为 **meen**,计算标准偏差的子函数为 **stdev**。函数和子函数存在于文件 *MeanStdDev.m* 中。下面是函数和子函数的实现代码:

```
function[m,s] = MeanStdDev(dat)          % Main function
n = length(dat);
m = meen(dat,n);
s = stdev(dat,n);

function m = meen(v,n)                   % Sub function
m = sum(v)/n;

function sd = stdev(v,n)                 % Sub function
m = meen(v,n);                           % Calls a sub function
sd = sqrt((sum(v.^2) - n * m^2)/(n - 1));
```

调用函数和子函数的程序为:

```
v = [1,2,3,4];
[m,s] = MeanStdDev(v)
```

执行时得到:

```
m =
  2.5000
s =
  1.2910
```

应该注意 **meen** 和 **stdev** 子函数不能独立使用。例如,在命令窗口中输入:

```
v = [1,2,3,4];
m = meen(v,length(v))
```

将产生错误信息。

如果可以确定为支持程序而创建的函数不会用于相邻上下文之外,则可以将程序转换为函数,而将函数文件内的所有函数转换为子函数。主函数的一般形式大都与 5.2.2 节中的特殊情况 3 类似。在 5.5.4 节中将讲述这个过程。

5.3 用户自定义函数、函数句柄和 feval

有许多 MATLAB 内置函数要求用户通过它们创建特定形式的用户自定义函数。这类函数使用 MATLAB 函数:

```
feval(FunctionHandle,p1,p2,...,pn)
```

其中，*FunctionHandle* 是函数句柄，后面会加以解释。$p1,p2,\cdots$是传递给 *FunctionHandle* 所指函数的参数。5.5节给出几个使用函数句柄和 feval 的 MATLAB 内置函数的例子。除此之外，有些情况下，用户也想具备这种功能。

在介绍使用 feval 的程序例子之前，首先介绍函数句柄。函数句柄是一种引用函数的方法，可以作为函数参数被传递，及使用 feval 计算。函数句柄由@和函数名构成。在下面的例子中将加以介绍。

在例4.8的结果基础上介绍其过程。例子中的求根程序确定了某函数 $f(x)=0$ 的 m 个最接近的根。将程序重新定义为 **ManyZeros** 函数，包含在 *ManyZeros.m* 文件中。函数 $f(x)$ 及其名字可以是任意的。此外，还需要定义 $f(x)$ 能输入多个参数。求根程序要求4个输入：m 为所希望的最接近根的数目，x_s 为根的初始值，t 确定接近零点 $f(x_{root})$ 程度的计算误差，Δ 为初始增量。

对于此例，定义：

$$f(x) = \cos(\beta x) - \alpha \qquad \alpha \leqslant 1$$

因而，必须赋给用户定义的函数两个值：β 和 α。函数名为 **CosBeta**，在文件 *CosBeta.m* 中。函数 *ManyZeros* 如下所示(参考例4.8)：

```
function nRoots = ManyZeros(zname,n,xs,toler,dxx,b,a)
x = xs;
dx = dxx;
for m = 1:n
  s1 = sign(feval(zname,x,b,a));
  while dx/x > toler
    if S1 ~ = sign(feval(zname,x + dx,b,a))
      dx = dx/2;
    else
      x = x + dx;
    end
  end
  nRoots(m) = x;
  dx = dxx;
  x = 1.05 * x;
end
```

函数 **CosBeta** 是：

```
function d = CosBeta(x,beta,alpha)
d = cos(x * beta) - alpha;
```

函数 **ManyZeros** 要求下列输入变量：定义 $f(x)$ 的函数名，函数句柄@**CosBeta** 和6个参数，前4个对应于 m，x_s，t 和 Δ，其余的2个是 β 和 α，这些都传递到函数 **CosBeta**。根据 m 的值，结果 c 可能为一标量($m=1$)或为一 m 阶向量。

为访问函数 **CosBeta** 并返回其数值，使用 MATLAB 函数 feval。MATLAB 函数 sign 用来判断 feval 返回数值的符号。要注意，在程序文件中定义的函数输入变量名和两个函数文件中定义的可能完全不同。这说明，由于函数中的变量是局部的，所以起作用的是参数的位置和用途，而不是名称的字符描述。

使用这些函数的程序为：

```
NoRoots = 5;xStart = 0.2;
tolerance = 1e - 6;increment = 0.3;
beta = pi/3;a = 0.5;
c = ManyZeros(@CosBeta, NoRoots,xStart,tolerance,increment,beta,a)
```

执行结果显示在命令窗口中：

```
C =
  1.0000  5.0000  7.0000  11.0000  13.0000
```

5.4　以数组变量为输入参数的 MATLAB 函数

5.4.1　概述

MATLAB 中有许多常用的函数,广泛用于以数值方法求解工程问题。下面考虑几个这样的函数。将其分为两组,一组以数组为输入参数,另一组需要用户自定义函数。本节讲述输入为数组的函数,5.5 节讲述需要用户自定义函数的部分。

本节要讲述的函数有：

polyfit	根据数组值进行多项式拟合
polyval	由已知数组值计算多项式值
spline	对数组数据应用三次样条函数插值
interp1	在两数间插值
trapz	由振幅值数组估算积分
fft/ifft	由采样数据求傅里叶变换及其反变换

5.4.2　多项式拟合——polyfit/polyval

若多项式的一般形式如下：

$$y(x) = c_1 x^n + c_2 x^{n-1} + \cdots + c_n x + c_{n+1} \tag{5.1}$$

其中,x 是输入值,y 是相应的输出。系数 c_k 由下面的语句决定。

```
c = polyfit(x,y,n)
```

其中,n 是多项式的阶次,$c = [c_1 c_2 \cdots c_n c_{n+1}]$是长度为$(n + 1)$的向量,代表式(5.1)中多项式的系数。$x$ 和 y 都是长度为 $m \geq n+1$ 的向量,并且是多项式拟合的数据,x 为输入,y 为输出。

求得 c 后,用下列语句可求出式(5.1)。

```
y = polyval(c,xnew)
```

其中,c 是长度为$(n+1)$的向量,并且由 polyfit 确定。x 是某一点的标量或向量表示,在其上对多项式进行计算。一般情况下,polyval 中的 *xnew* 值可以任意选择,可以与 x 值相同也可以不同。

下面用例子说明这些函数的使用。

例 5.1　用于钢槽灵敏度计算的 Neuber 常数

金属槽的敏感系数 q 可由 Neuber 常数$\sqrt{a}$和槽口半径 r 定义,表达式如下：

$$q = \left(1 + \frac{\sqrt{a}}{\sqrt{r}}\right)^{-1}$$

$\sqrt{a}$的值对于不同的金属是不同的，并且是材料极限强度 S_u的函数。给定某金属，可通过拟合的方法，利用多项式近似得到$\sqrt{a}$的数值。一旦有了这个多项式，给定 r 和 S_u 的值即可确定 q 的值。

表 5.2 钢的 Neuber 常数

S_u(ksi)	$\sqrt{a}(\sqrt{in})$	S_u(ksi)	$\sqrt{a}(\sqrt{in})$
50	0.130	170	0.028
70	0.092	190	0.020
90	0.072	210	0.015
110	0.057	230	0.010
130	0.046	250	0.007
150	0.037		

为了对表 5.2 中所列数据进行拟合，并求取$\sqrt{a}$相对于 S_u($50 \leqslant S_u \leqslant 250$ ksi)和 r($0 < r < 0.2$ in) 的值编写程序。为了简化，假定每次输入一组 S_u和 r。进而，将表 5.2 中的数据放在名为 **NeuberData** 的函数中。图 6.20(b)给出了一系列曲线，表示 q 值如何随数据变化。

该数据函数如下：

```
function nd = NeuberData
nd = [50,.13;70,.092;90,.072;110,.057;130,.046;150,.037;...
    170,.028;190,.020;210,.015;230,.010;250,.007];
```

其中，$nd(:,1) = S_u$，$nd(:,2) = \sqrt{a}$。程序如下：

```
ncs = NeuberData;
c = polyfit(ncs(:,1),ncs(:,2),4);
r = input('Enter notch radius(0 < r < 0.2 in.)');
Su = input('Enter ultimate strength of steel(50 < Su < 250 ksi)');
q = 1/(1 + polyval(c,Su)/sqrt(r));
disp(['Notch sensitivity = ' num2str(q)])
```

执行这一程序可得到以下结果：

```
Enter notch radius(0 < r < 0.2 in.)0.1
Enter ultimate strength of steel(50 < Su < 250 ksi) 135
Notch sensitivity = 0.87999
```

其中，在程序前两行用户输入 0.1 和 135 两个数，边输入边显示。然后，计算 q 的值并在第三行显示。

5.4.3 用 spline 实现数据拟合

由一组离散数值产生光滑曲线的较好方法是使用 splines。产生曲线的函数形式为：

```
Y = spline(x,y,X)
```

其中，y 是$y(x)$，x 和 y 是同样长度的向量，用于产生函数关系 $y(x)$。X 是一标量或是 $Y = y(X)$ 需要的向量，一般 $x \neq X$。下面说明 spline 的用法。

例 5.2 数据拟合为指数衰减正弦波

首先,产生一些指数衰减振荡数据,然后用一系列 splines 对这些数据拟合。给定衰减系数 $\xi<1.0$,对以下函数在时间 τ 上采样,产生数据:

$$f(\tau,\xi)=\frac{e^{-\xi\tau}}{\cos\alpha}\cos\left(\tau\sqrt{1-\xi^2}+\alpha\right) \tag{5.2}$$

其中,

$$\alpha=\tan^{-1}\frac{-\xi}{\sqrt{1-\xi^2}}$$

在函数 **DampedSineWave** 中计算表达式,程序如下:

```
function f = DampedSineWave(tau,xi)
alpha = atan( - xi/sqrt(1 - xi^2));
f = exp( - xi * tau). * cos(tau * sqrt(1 - xi^2) + alpha)/cos(alpha);
```

在 $0\leqslant\tau\leqslant20$ 范围内采样 $f(\tau,\xi)$ 的 15 个等间隔点,用 200 个 τ 的等间隔值画出多项式的一段波形。将 $\xi=0.1$ 的原始波形与拟合后的结果多项式画在一起进行比较。程序如下:

```
n = 10;xi = 0.1;
tau = linspace(0,20,n);
data = DampedSineWave(tau,xi);
newtau = linspace(0,20,200);
plot(newdata,spline(tau,data,newtau),'k--',...
        newtau,DampedSineWave(newtau,xi),'k-')
```

程序执行后产生的图形如图 5.1 所示,虚线为拟合数据。可以看出结果相当理想,原始曲线与拟合曲线变得几乎无法区分。对 plot 的详细讨论参见 6.2 节。使用 polyfit 在同样范围内不能得到如此光滑的拟合曲线。

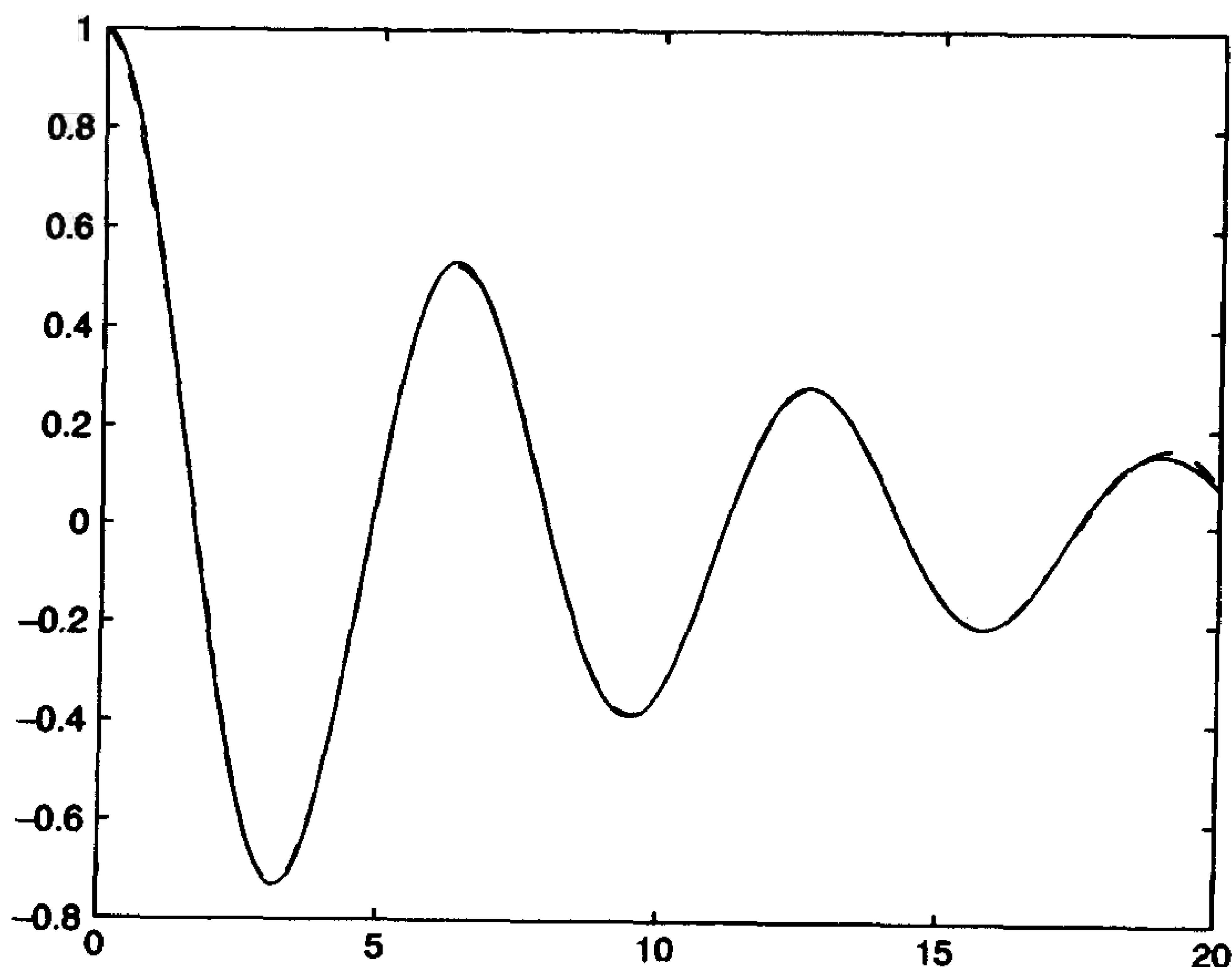


图 5.1 衰减正弦波(实线)与在 $0\leqslant\tau\leqslant20$ 范围内 15 等间隔点采样所获得的近似值波形(虚线)的比较

5.4.4 插值——interp1

为估计出位于两点间某一点的近似位置，必须采用插值计算。实现插值的函数为：

```
V = interp1(u,v,U)
```

其中，v 是 $v(u)$，u 和 v 是同阶次的向量，$\boldsymbol{U}$ 是一标量，或者是一个由 $\boldsymbol{V}$ 所确定的一系列 u 值构成的向量。$\boldsymbol{V}$ 和 $\boldsymbol{U}$ 的阶次相同，一般 $u \neq U$。下面解释 interp1 的使用。

例 5.3 指数衰减正弦波的第一个0点

由式(5.2)产生数据集进行指数衰减正弦波的曲线拟合，用函数 **DampedSineWave** 加以实现。我们感兴趣的是从这些数据点中估计出第一个0点。由图5.1可以看出，0点出现在 $\tau=4.5$ 之前，$\tau>4.5$ 后，就非常接近第二个0点。在 $0\leqslant\tau\leqslant4$ 范围内产生指数衰减正弦波的15对数据值，使用 interp1 估计出使 $f(\tau,\zeta)\approx0$ 的 τ 值，这里假定 $\zeta=0.1$。程序为：

```
xi = 0.1;
tau = linspace(0,4.5,15);
data = DampedSineWave(tau,xi);
TauZero = interp1(data,tau,0)
```

执行结果为：

```
TauZero =
  1.6817
```

由式(5.2)得到的精确答案为：

$$\tau=\frac{\pi/2-\alpha}{\sqrt{1-\xi^2}}=\frac{1}{\sqrt{1-\xi^2}}\left(\pi/2-\tan^{-1}\frac{-\xi}{\sqrt{1-\xi^2}}\right)=1.6794$$

5.4.5 数值积分函数——trapz

有几种近似求简单积分的方法。本节首先介绍函数 trapz，其输入参数要求以数组的形式给出。另一函数 quadl 要求用积函数以函数的形式给出，在5.5.2节将会加以介绍。5.6节将介绍一种符号技术求积分。

首先，介绍下列函数，

```
Area = trapz(x,y)
```

在此方式下，可以用数组的形式指定 x 的值和对应的 y 值。函数执行以下运算：首先将 x 分成小的区间，利用相应 y 的平均值与 x 区间相乘，再相加。

下面介绍此函数的使用。在下例中，用到函数 diff。diff 可以计算向量相邻元素之间的差值。也就是，对向量 $\boldsymbol{x}=[x_1 x_2 \cdots x_n]$，可以计算出 $(n-1)$ 维向量 $\boldsymbol{q}$，见下式：

$$\boldsymbol{q}=[x_2-x_1, x_3-x_2, \cdots, x_n-x_{n-1}]$$

对向量 $\boldsymbol{x}$ 而言，diff 简写为：

```
q = x(2:end) - x(1:end - 1);
```

例5.5显示了怎样用 diff 求空间直线的长度。

例 5.4 指数衰减正弦波曲线面积

应用 trapz 函数,求式(5.2)和图 5.1 表示的指数衰减正弦波关于 x 轴的有效面积,以及在$0 \leqslant \tau \leqslant 20$ 范围内正负区域对有效面积的贡献。

下面的程序实现在 $\xi = 0.1$ 时根据 200 个数据点计算阻尼正弦波的面积。

```
xi = 0.1; N = 200;
tau = linspace(0,20,N);
ftau = DampedSineWave(tau,xi);
Area = trapz(tau,ftau)
```

执行程序得:

```
Area =
  0.3021
```

求正弦波正负区域的面积比较复杂。与求总面积的过程类似。首先,创建 200 个等间隔 τ 值,用函数 **DampedSineWave** 计算 $\xi = 0.1$ 时相应的 $f(\tau,\xi)$。正负区域由 x 轴来划分,用 find 和 diff 函数求出 $f(\tau,\xi)$为正或负的不同部分的索引,然后使用索引分离出不同区域。同时,使用 diff 和 find 函数以求出每个正区域或负区域的起始和终止位置。当 $f(\tau,\xi)$ 由正变为负然后又变为正时,由 find 函数求得的索引值会有一个空隙。这样在正区域内相邻两个索引之间的差值为 1,而遇到负区域时索引值会有一个空隙。当曲线又回到正区域时,索引差值又会大于 1。图 5.1 的指数衰减正弦波是从正区域开始的,且曲线末尾位于正区域内。根据上述考虑,编写程序如下所示:

```
xi = 0.1; N = 200;
tau = linspace(0,20,N);
ftau = DampedSineWave(tau,xi);
% Positive area
indx = find(ftau >= 0);
reg = find(diff(indx) > 1);
reg = [1 reg length(indx)];
L = length(reg);
PosArea = [];
% Negative area
indxn = find(ftau < 0);
regn = find(diff(indxn) > 1);
regn = [1 regn length(indxn)];
NegArea = [];
for k = 1:L - 1
  % Positive Area
  if k == 1
    idx = indx(reg(k):reg(k + 1));
  else
    idx = indx(reg(k): + 1;reg(k + 1));
  end
  PosArea = [PosArea trapz(tau(idx),ftau(idx))];
  % Negative area
  if k == 1
    idxn = indxn(regn(k):regn(k + 1));
    NegArea = [NagArea trapz(tau(idxn),ftau(idxn))];
  elseif k < L - 1
```

```
        idxn = indxn(regn(k) + 1;regn(k + 1));
        NegArea = [NagArea trapz(tau(idxn),ftau(idxn))];
    end
end
TotalPosArea = sum(PosArea)
TotalNegArea = sum(NegArea)
TotalArea = TotalPosArea + TotalNegArea
```

执行程序,得:

```
TotalPosArea =
  2.9473
TotalNegArea =
  -2.6472
TotalArea =
  0.3001
```

由此看出总面积与前面所求非常接近。

例5.5　空间直线的长度

空间中一直线段的长度可以由下列积分公式求得:

$$L = \int_a^b \sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}t}\right)^2}\,\mathrm{d}t \approx \sum_{i=1}^{N} \sqrt{(\Delta_i x)^2 + (\Delta_i y)^2 + (\Delta_i z)^2}$$

其中,

$$\Delta_i x = x(t_{i+1}) - x(t_i)$$
$$\Delta_i y = y(t_{i+1}) - y(t_i)$$
$$\Delta_i z = z(t_{i+1}) - z(t_i)$$

并且,$t_1 = a, t_{N+1} = b$。

为了进一步说明 L 的近似求解过程,令:

$$x = 2t$$
$$y = t^2$$
$$z = \ln t$$

t 满足 $1 \leqslant t \leqslant 2$,且 $N = 25$,参量 $\Delta_i x, \Delta_i y$ 和 $\Delta_i z$ 可由 diff 计算得出。程序如下:

```
t = linspace(1,2,25);
L = sum(sqrt(diff(2 * t).^2 + diff(t.^2).^2 + diff(log(t)).^2))
```

执行结果为:

```
L =
  3.6931
```

5.4.6　数字信号处理——fft 和 ifft

离散傅里叶变换

在 $0 \leqslant t \leqslant T$ 内每隔 Δt 时间对实函数 $g(t)$ 进行一次采样,则实函数 $g(t)$ 的傅里叶变换可由离散傅里叶变换近似表示为:

$$G_n = G(n\Delta f) = \Delta t \sum_{k=0}^{N-1} g_k \mathrm{e}^{-j2\pi nk/N} \qquad n = 0, 1, \cdots, N-1$$

其中，$g_k = g(k\Delta t)$，$\Delta f = 1/T$，$T = N\Delta t$，N 是采样数，参见图 5.2。一般而言，G_n 是一复数。对 Δt 的限制是：

$$\alpha \Delta t < \frac{1}{f_h}$$

其中，f_h 是 $g(t)$ 中的最高频率分量，并且 $\alpha \geqslant 2$。量 G_n 称为 $g(t)$ 的幅度密度，其单位是"幅度 - 秒"，或等价为"增益/Hz"。

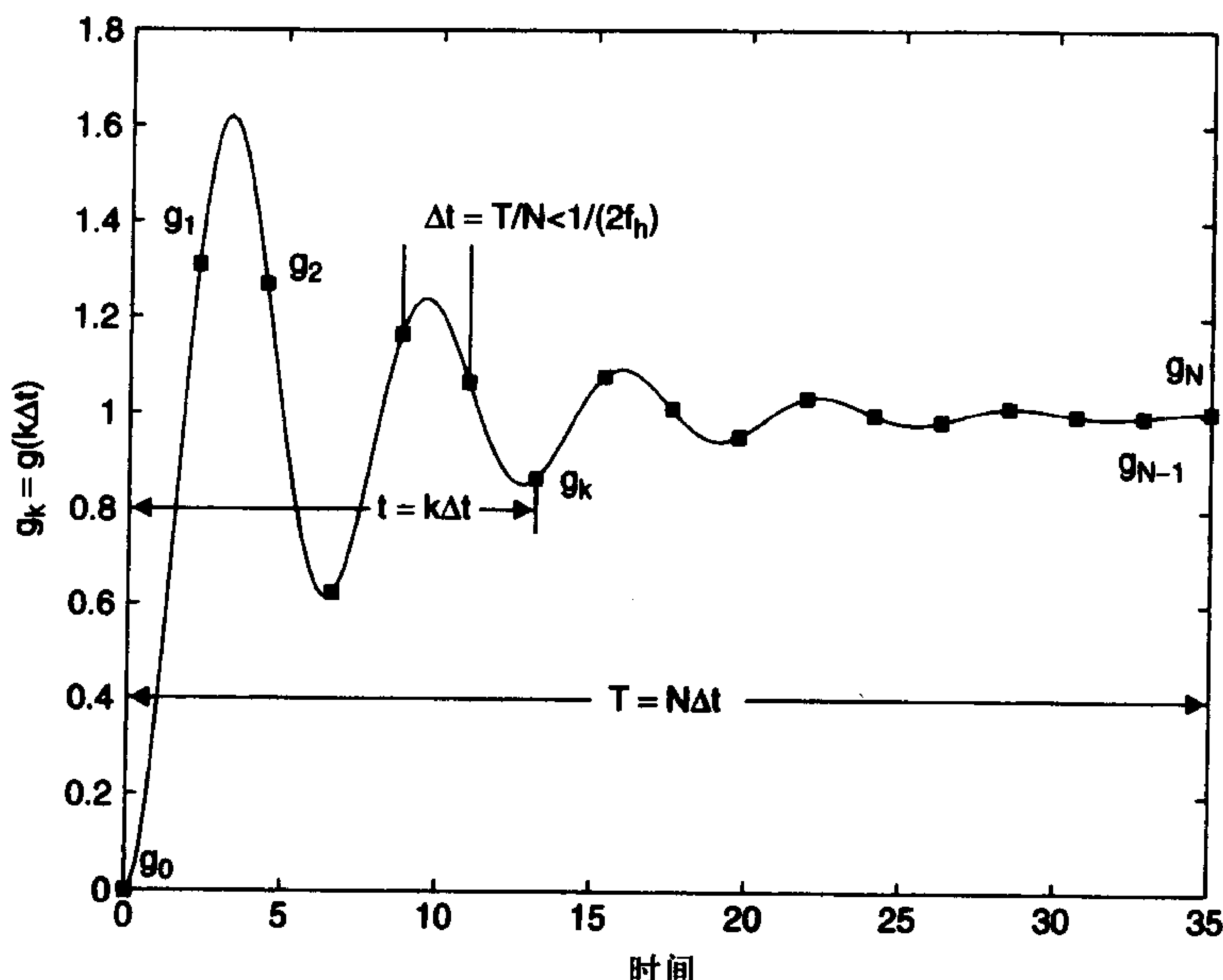


图 5.2 采样波形

反变换由下式近似表示：

$$g_k = \Delta f \sum_{n=0}^{N-1} G_n e^{j2\pi nk/N} \qquad k = 0,1,\cdots,N-1$$

为估计出对应于每一个 G_n 在频率 $n\Delta f$ 处的 A_n 值，应计算 G_n 与 Δf 的乘积，即：

$$A_n = \Delta f G_n$$

所以：

$$A_n = \frac{1}{N} \sum_{k=0}^{N-1} g_k e^{-j2\pi nk/N} \qquad n = 0,1,\cdots,N-1$$

因为 $\Delta f \Delta t = 1/N$，所以信号的平均功率为：

$$P_{avg} = \sum_{n=0}^{N-1} |A_n|^2$$

$|A_n|$ 经常作为 $n\Delta f$ 的函数，以得到幅度频谱。此时为①：

$$|A_n|_s = 2|A_n| \qquad n = 0,1,\cdots,N/2-1$$

这些表达式最好用快速傅里叶变换(FFT，fast Fourier transform)计算，而 FFT 是数值计算离散傅里叶变换很有效的算法。当采样数据点数是 2 的整次幂，即 $N = 2^m$ 时较好。其中，m 是

① 参见 J. S. Bendat 和 A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*, John Wiley & Sons, New York, 1980 年。

正整数。FFT 算法用以下函数实现

```
G = fft(g,N)
```

反变换为：

```
g = ifft(G,N)
```

其中，$G = G_n/\Delta t$，$g = g_k/\Delta f$。

权函数

在很多种情况下，通过某一适当的函数对 $g(t)$ 加权，以便在另一个变换域中提供更好的解或其他属性。在进行离散傅里叶变换之前，先调整原始信号，以便消除因窗口函数变化对信号平均值和信号平均功率的影响。如果权函数为 $w_n = w(n\Delta t)$，则被修正信号 g_{cn} 由下式给出[①]：

$$g_{cn} = k_2 w_n (g_n - k_1) \qquad n = 0,1,\cdots,N-1$$

其中，

$$k_1 = \sum_{n=0}^{N-1} w_n g_n \Big/ \sum_{n=0}^{N-1} w_n$$

对窗口函数的平均值进行修正，并且：

$$k_2 = \left[N \Big/ \sum_{n=0}^{N-1} w_n^2 \right]^{1/2}$$

对窗口函数的平均功率进行修正。然后，对 g_{cn} 进行离散傅里叶变换。

在 MATLAB 的数字信号处理工具箱中包含 8 个常用的权函数。

互相关

两个有限宽的确知信号 $x(t)$ 和 $y(t)$ 的互相关系由下式给出：

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t) y(t+\tau) \mathrm{d}t \qquad -\infty < \tau < \infty$$

对互密度谱函数 $S_{xy}(\omega)$ 进行傅里叶反变换可估计出这一量：

$$R_{xy}(\tau) = F^{-1}[S_{xy}(\omega)]$$

其中，$F^{-1}(\cdots)$ 表示傅里叶反变换，并且：

$$S_{xy}(\omega) = X(\omega) Y^*(\omega)$$

$X(\omega)$ 和 $Y(\omega)$ 分别是 $x(t)$ 和 $y(t)$ 的傅里叶变换。星号表示复共轭。为将 $R_{xy}(\tau)$ 变换为正确的单位，要求 $S_{xy}(\omega)$ 乘以 $\Delta t = T/N$。

用两个例子说明这些联系。

例 5.6 正弦波的傅里叶变换

对周期为 T 的正弦波采样，

令：

$$g(t) = A_o \sin(2\pi f_o t) \qquad 0 \leqslant t \leqslant T = 2^K/f_o \qquad K = 0,1,2,\cdots$$

而且，

$$\Delta t < 1/(2f_o) \text{ 或 } m - K > 1$$

① 参见 E. C. Ifeachor 和 B. W. Jervis, *Digital Signal Processing: A Practical Approach*, Addison-Wesley, Harlow, England, 1993 年，第 593 页。

因为：

$$f_h = f_o = 2^K/T$$

且，

$$\Delta t = 2^{-m}T$$

对 $g(t)$ 用下面的汉明(Hamming)函数加权：

$$w(t) = 0.54 - 0.46\cos(2\pi t/T) \qquad 0 \leqslant t \leqslant T$$
$$= 0 \qquad 其他$$

下面的程序具有的功能是计算和绘制校正后的信号 $g_c(t)$，及其幅度谱 A_n，显示信号的平均功率。在此，$P_{avg} = A_o^2/2$。设 $A_o = 2.5, f_o = 10$ Hz, $K = 5, m = 10 (N = 1024)$。则：

```
k = 5;m = 10;fo = 10;Ao = 2.5;
N = 2^m;T = 2^k/fo;
ts = (0:N - 1)' T/N;
df = (0:N/2 - 1)/T;
whamm = 0.54 - 0.46 * cos(2 * pi * ts/T);
SampledSignal = Ao * sin(2 * pi * fo * ts);
k1 = sum(whamm. * SampledSignal)/sum(whamm);
k2 = sqrt(N/sum(whamm.^2));
CorrectedSignal = whamm. * (SampledSignal - k1) * k2;
figure(1)
plot(ts,CorrectedSignal)
figure(2)
An = abs(fft(CorrectedSignal,N))/N;
plot(df,2 * An(1:N/2))
disp(['Average power = ' num2str(sum(An.^2))])
```

执行程序得到图 5.3 和图 5.4，并且在 MATLAB 命令窗口中得到以下信息：

```
Average power = 3.125
```

MATLAB 函数 figure 用于提供两个独立的窗口外形，参见 6.1 节。注意，在图 5.4 中正弦波幅值不等于2.5。当权函数被除去时，可得到真正的幅值。

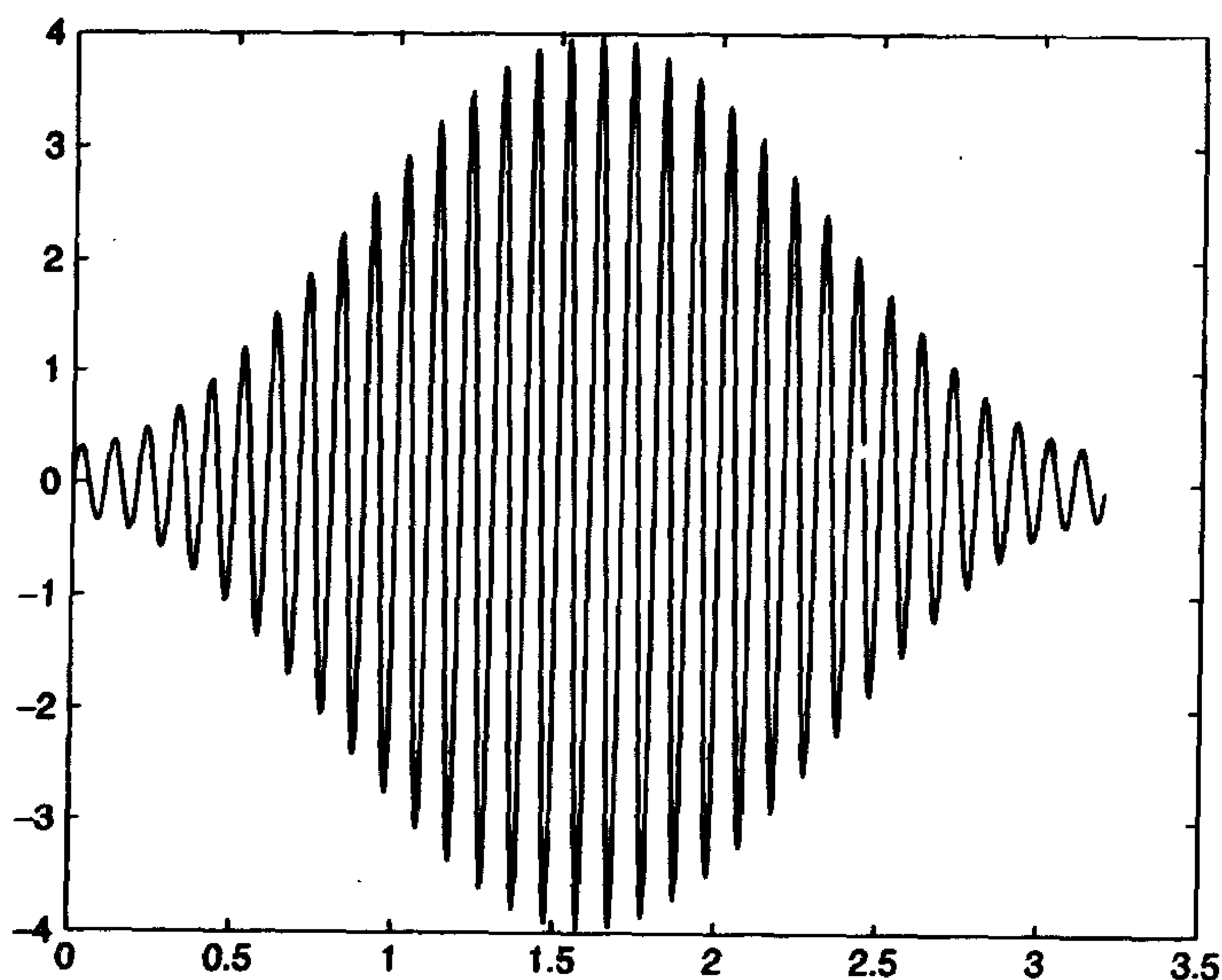


图 5.3 汉明权函数修正的正弦波

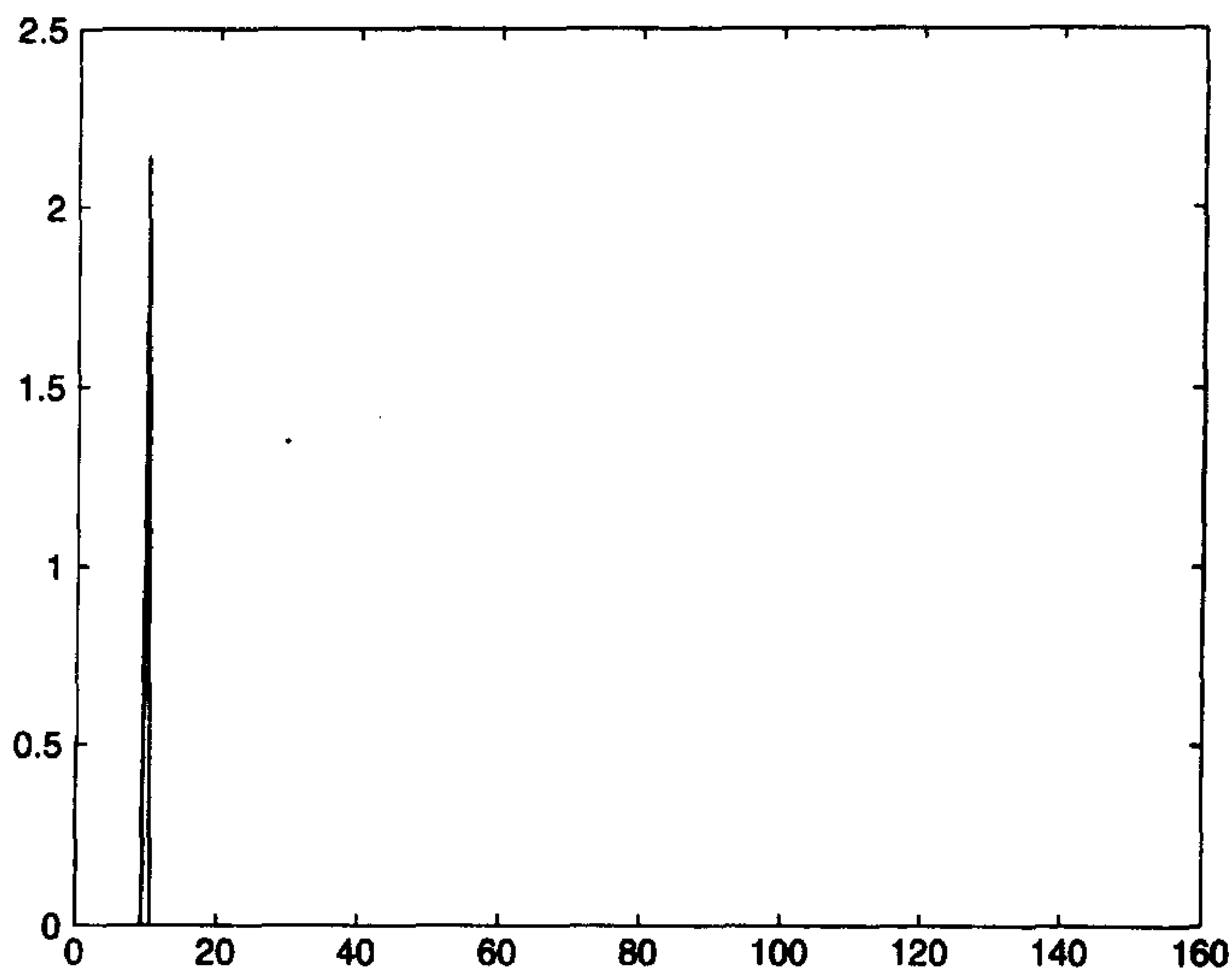


图5.4 应用汉明权函数的正弦波幅度谱

例5.7 两脉冲的互相关

对于图5.5所示的两个脉冲,互相关函数的表达式如下:

$$x(t) = A_x[u(t) - u(t - T_o)] \qquad t \geqslant 0$$
$$y(t) = A_y[u(t - T_1) - u(t - T_1 - T_2)] \quad t \geqslant 0$$

其中,$u(t)$为单位阶跃函数。设 $A_x = A_y = 1, T_o = 0.01$ s, $T_1 = 2T_o, T_2 = T_1 + T_o, T = T_2 + T_o$, $N = 2^{10}$。写出以下程序:

```
To = 0.01;T1 = 2 * To;T2 = T1 + To;Tend = T2 + To;
N = 2^10;deltaT = Tend/N;Ax = 1;Ay = 1;
t = linspace(0,Tend,N);
PulseCrossCorr = inline('ampl * ((t - Ts >= 0) - (t - Te > 0))','t','Ts','Te','ampl');
x = PulseCrossCorr(t,0,To,Ax);
y = PulseCrossCorr(t,T1,T2,Ay);
X = fft(x,N);
Y = conj(fft(y,N));
Rxy = ifft(X. * Y * deltaT,N);
plot(t,real(Rxy))
```

程序的执行结果参见图5.6。由于数值计算会带来截断误差,所以函数 real 用来除去残留的虚部。

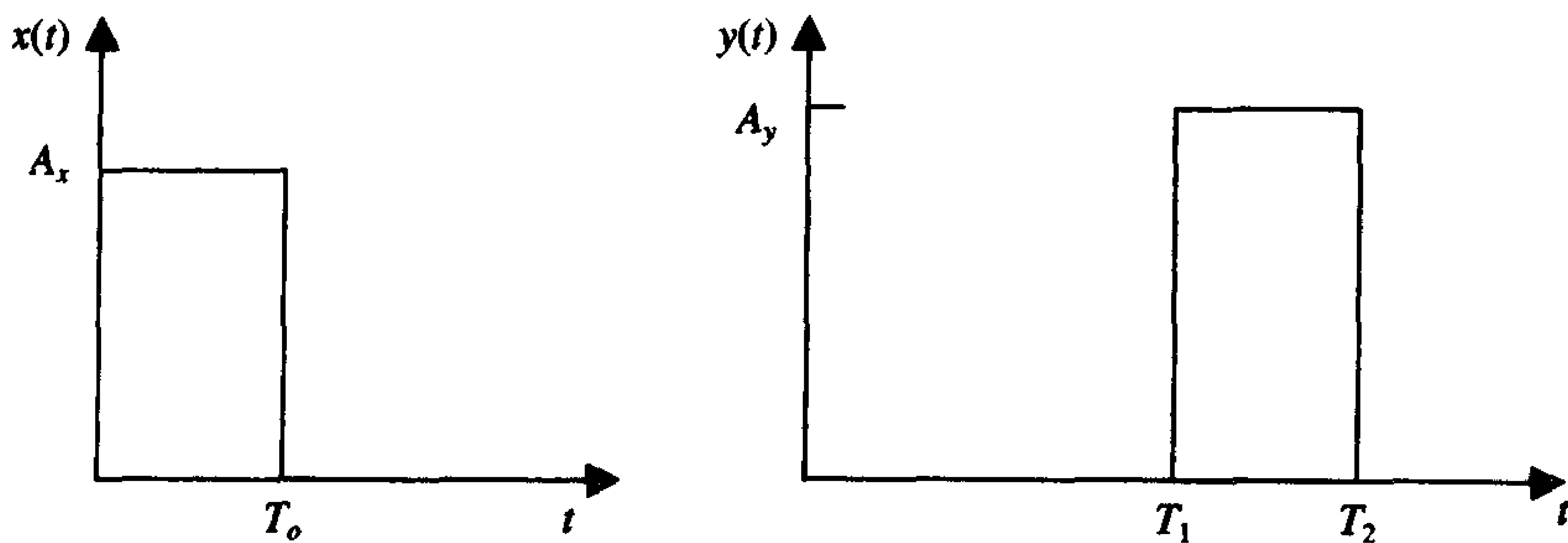


图5.5 两脉冲

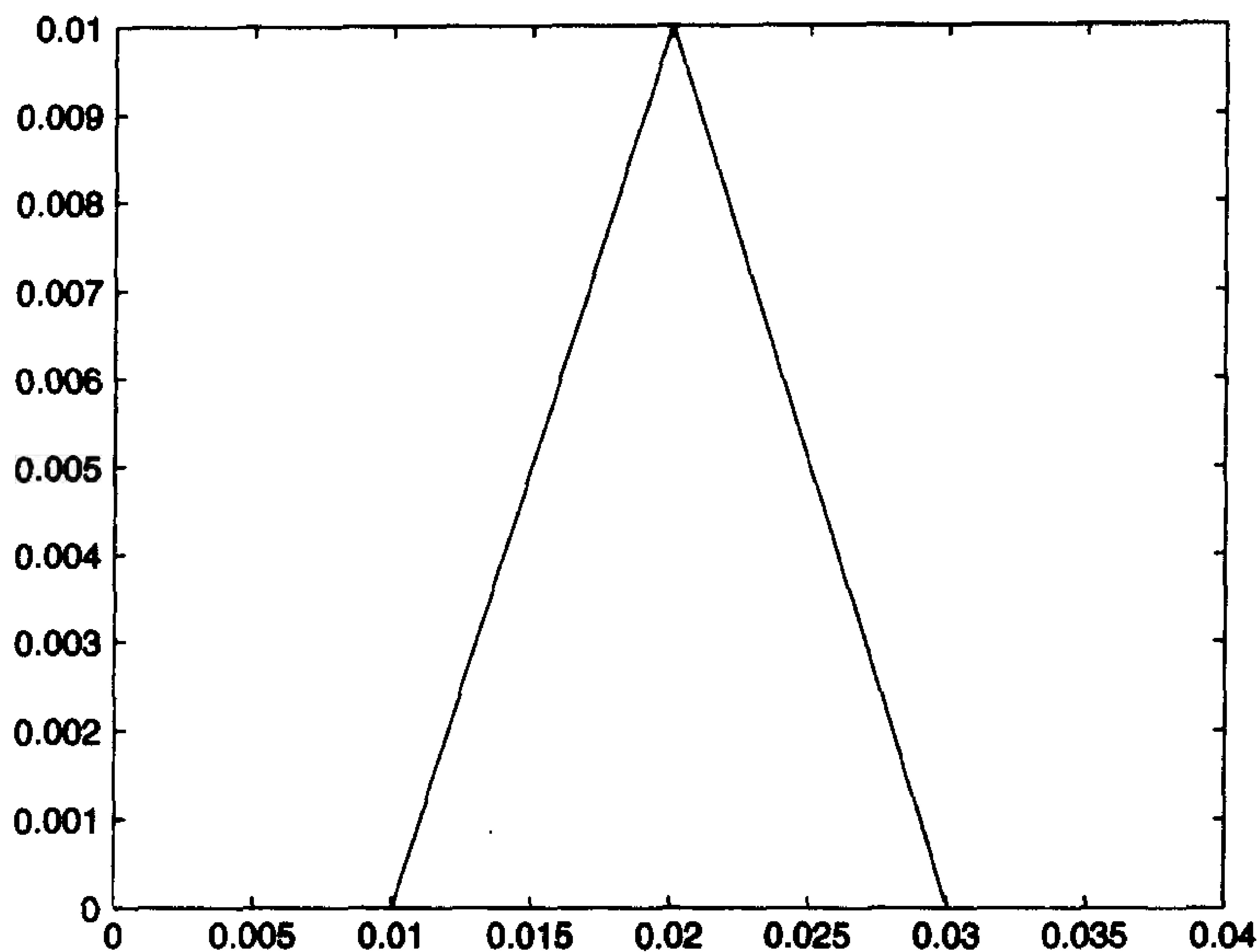


图 5.6 两个等周期脉冲的互相关函数

5.5 使用用户定义函数的 MATLAB 函数

MATLAB 提供了几个函数用来计算用户定义函数,分别是:

fzero——找 $f(x)=0$ 的根
roots——找一多项式的根
quadl——在给定间隔内对 $f(x)$的数值积分
dblquad——在给定间隔内对 $f(x,y)$积分
ode45——给定初始条件求常微分方程组的解
bvp4c——给定边界条件时求常微分方程组
fminbnd——在一特定间隔内找到 $f(x)$的局部最小值
fsolve——非线性方程组的数值解

最后一个函数 fsolve 来自优化工具箱。

使用内置函数时,用户自定义函数的输入参数或输出结果应满足某种特定的要求。下面几节说明了这些不同的要求,并且在函数的帮助文件中清楚地表示出来。

5.5.1 零点函数——fzero 和 roots/poly

对于许多方程找不到精确的代数方法求 $f(x)=0$ 的解,而需要用数值方法求解。容许误差为 t_o时,在 x_o的相临区域内或在$[x_1,x_2]$范围内,利用内置函数 fzero 可以求解$f(x)=0$,当 $j=1,2,\cdots$时,该函数也可以将 p_j参数传递到函数定义$f(x)$中。$f(x)$必须在间隔$[x_1,x_2]$内变化,否则会出错。一般的表达式为:

```
z = fzero(@FunctionName,x0,options,p1,p2,...)
```

其中,z 是满足$f(z)\approx 0$ 的值,**FunctionName** 是函数文件名,但是没有后缀“.m”。x0 = x_o或 x0 = $[x_1\ x_2]$, p1, p2,等等,都是参数 p_j。此后用 optimset 设置 *options*。

内置函数 optimset 是一通用参数调整函数，由优化工具箱中的 MATLAB 函数使用，最简单的例子是使用 optimset 关闭显示，如下所示：

```
options = optimset('Display','off');
```

上式将会用在每个 fzero 语句之前。通过 optimset 的帮助文件，可知其属性会根据函数变化而变化。

函数接口具有以下形式：

```
function z = FunctionName(x,p1,p2,...)
Expression(s)
```

其中，x 是独立变量，为了找到使 $f(x=z)\cong 0$ 的值，fzero 不断改变 x 的值。*Expressions* 必须能处理 x 向量，因此 *Expressions* 通常用点标记表示。独立变量必须总出现在这一位置。用 MATLAB 创建的大多数用于计算的函数应满足这一要求，本章所述的函数也都如此。fzero 还可以与 inline 一起使用，其用法如下：

```
InlineFunctionName = inline('Expression','x','p1','p2',...);
z = fzero(InlineFunctionName,x0,options,p1,p2,...)
```

或者：

```
z = fzero(inline('Expression','x','p1','p2',...),x0,options,p1,p2,...)
```

只有当 *Expression* 的形式非常简单时，才使用第一种方式。

下面说明使用 fzero 可以避免得到很差的 $x0$。函数 $f(x)$ 可以是 MATLAB 函数也可以是用户创建的函数。假定要求 $\cos(x)$ 在 $x=2\pi$ 附近的根，选 $x0=2\pi$ 写出以下语句：

```
options = optimset('Display','off');
w = fzero(@cos,2*pi,options)/pi
```

结果：

```
w =
   1.5000
```

也就是 $\cos(1.5\pi)=0$。然而，修改 $x0=2.04\pi$ 时：

```
options = optimset('Display','off');
w = fzero(@cos,2.04*pi,options)/pi
```

则结果：

```
w =
   2.5000
```

如果 $x0=2.03\pi$，则：

```
options = optimset('Display','off');
w = fzero(@cos,2.03*pi,options)/pi
```

得：

```
w =
   1.5000
```

对于多值函数，应使用 $x0=[x_1\ x_2]$ 形式，并且准确划定范围。然而，如果 $f(x_1)$ 的符号和 $f(x_2)$ 的符号相同，则会得到一个错误结果。所以，下式将会导致一个错误：

```
options = optimset('Display','off');
```

```
w=fzero(@cos,[0, 2*pi],options)/pi
```

而式:

```
options=optimset('Display','off');
w=fzero(@cos,[0.6*pi, 2*pi],options)/pi
```

得出:

```
w =
   1.5000
```

对多值函数而言,其特征是不分先后,所以应当首先确定根的大致位置。

另一方面,如果对方程 $J_1(x)=0$[其中 $J_1(x)$是一类1阶贝塞尔函数[①]]在3附近的根感兴趣,则以下语句将得不到预想的结果。

```
besselj(n,x)
```

其中,n 是阶次(本例中为1),x 是独立变量。由于第一个参数是 n 而不是 x,所以不能直接使用该函数,而是用 inline 函数创建一个新的函数:

```
besseljx=inline('besselj(n,x)','x','n');
options=optimset('Display','off');
a=fzero(besseljx,3,options,1)
```

返回正确的结果:

```
a =
   3.8317
```

注意,为将参数 $p_1=n=1$ 传递给函数 *besseljx*,在 fzero 的第四个位置必须赋以数值1。

下面再举几个使用 fzero 的例子。

例5.8 两端受力梁的5个最低固有频率系数

一根两端受力的梁,其固有频率系数由下式确定(参见9.4.1节):

$$f(\Omega) = \cos(\Omega)\cosh(\Omega) - 1$$

现在,确定下式的最小的5个根(不包括 $\Omega=0$)。该过程首先在 Ω 范围内画出函数,以观察在哪个位置上的 Ω 使 $f(\Omega)=0$。利用这些信息可以得到5个范围,fzero 可在每个范围上得到一个根。程序如下:

```
qcc=inline('cos(x).*cosh(x)-1','x');
options=optimset('Display','off');
% Om=linspace(0,20);
% plot(Om,qcc(Om))
% axis([0,20,-10,10])
xo=[3,5];
for n=1:5
  q(n)=fzero(qcc,xo,options);
  xo=[1.05*q(n),q(n)+4];
end
disp(['Lowest five natural frequency coefficients are:')
disp(num2str(q))
```

---

① 参见 F.B. Hildebrand, *Advanced Calculus for Applications*, Prentice-Hall, Upper Saddle River, NJ, 1976年。

执行时,MATLAB 命令窗口显示以下内容:

```
Lowest five natural frequency coefficients are:
4.73004  7.8532  10.9956  14.1372  17.2788
```

因为 qcc 在很大的正负范围内变化,用 axis 限制显示值极大地提高了图形的求解能力。在这种情况下,限制 y 轴为 ±10,参见 6.2 节。从图中可以确定,Ω 的搜索范围应从前一个根的 1.05 倍处开始,到前一个根加 4 的位置,搜索范围的上限应略高于下一个零点。在程序的实际编制过程中,先让前 5 个表达式执行完毕,并对结果进行分析,然后再写出后 6 个表达式。一旦确认了这些信息,就可以将表达式注释掉。

如果 $f(\Omega)$ 不是一个常数(本例中为 4),而是要求有几个不同的值 C_j, $j = 1,2,3,4$。则上面的程序将变成以下形式:

```
qcc = inline('cos(x). * cosh(x) - 1','x');
options = optimset('Display','off');
C = [4,5,6,5,0];
xo = [3,5];
for n = 1:5
  q(n) = fzero(qcc,xo,options);
  xo = [1.05 * q(n),q(n) + C(n)];
end
disp('Lowest five natural frequency coefficients are:')
disp(num2str(q))
```

由于向量 C 的下标要求,所以必须增加第 5 个元素 0,但并不会使用 fzero 函数。

例 5.9　级数函数的零点

求满足下式的 a 值:

$$\sum_{j=1}^{1000} \frac{1}{j^2 - a} = 0$$

并在 MATLAB 窗口中显示结果。令初始值为 $\pi/2$,程序为:

```
suma = inline('sum(1./([1:1000].^2 - a))','a');
options = optimset('Display','off');
fofa = fzero(suma,pi/2,options);
disp(['The value of a is ' num2str(fofa)])
```

执行这些语句,可得以下结果:

```
The value of a is 2.0466
```

当 $f(x)$ 是具有下面形式的多项式时:

$$f(x) = c_1 x^n + c_2 x^{n-1} + \cdots + c_n x + c_{n+1}$$

用下面的函数可以很容易找到它的根:

```
roots(c)
```

其中,$c = [c_1, c_2, \cdots, c_{n+1}]$。例如,若:

$$f(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

则,

```
r = roots([1, - 10,35, - 50,24])
```

执行该语句得:

```
r =
   4.0000
   3.0000
   2.0000
   1.0000
```

roots 的逆为:

```
c = poly(r)
```

上式的返回值 c 是多项式的系数,r 是根向量。一般情况下,r 可以是实向量,也可以是复向量。因此,

```
r = roots([1, - 10,35, - 50,24]);
c = poly(r)
```

执行结果为:

```
c =
  1.0000   - 10.0000   35.0000   - 50.0000   24.0000
```

多项式可以用下式实现乘法运算:

```
conv(a,b)
```

其中,$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 是包含某个多项式系数的向量。另一多项式如下:

$$g(x) = x^2 - 4$$

则乘积 $h(x) = g(x)f(x)$通过下式得到:

```
h = conv([1,0, - 4],[1, - 10,35, - 50,24])
```

其结果为:

```
h =
  1   - 10   31   - 10   - 116   200   - 96
```

相应的多项式为:

$$h(x) = x^6 - 10x^5 + 31x^4 - 10x^3 - 116x^2 + 200x - 96$$

注意,根的结果按任意顺序排列。

5.5.2 数值积分函数——quad1 和 dblquad

内置函数 quad1 在容许误差 t_o内,对 $f(x)$从下限 a 到上限 b 进行数值积分。也可以将 p_j参数传递到定义$f(x)$的函数中去。quad1 的一般表达式为:

```
A = quad1(@FunctionName,a,b,t0,tc,p1,p2,…)
```

其中,**FunctionName** 是函数 M 文件的文件名(没有后缀".m"),$a = a$,$b = b$,$t0 = t_o$(省略时采用默认值),p1,p2 等是参数 p_j,当 $tc \neq [\,]$时,quad1 提供中间输出。函数由 inline 创建时,有:

```
A = quad1(InlineFunctionName,a,b,t0,tc,p1,p2,...)
```

其中，**InlineFunctionName** 是 inline 函数名。

函数 M 文件的接口形式为：

```
function z = FunctionName(x, p1, p2, ...)
Expression(s)
```

其中，x 是 quadl 的积分变量，并且必须总是出现在这一位置。inline 接口为：

```
InlineFunctionName = inline('Expression', 'x', 'p1', 'p2', ...)
```

下面说明 quadl 的使用方法。

例 5.10　求面积和重心

在机械学中经常令人感兴趣的两个量是二维图形的面积及其重心位置。假定两条曲线 $y_j = f_j(x), j = 1,2$，其交点为 x_1 和 x_2。在曲线两个交点之间的面积可用下式表示：

$$A = \int dA = \int_{x_1}^{x_2} (y_2 - y_1) dx$$

面积相对于原点的重心坐标分别为：

$$x_c = \frac{1}{A}\int x dA = \frac{1}{A}\int_{x_1}^{x_2} x(y_2 - y_1) dx$$

$$y_c = \frac{1}{A}\int y dA = \frac{1}{A}\int_{x_1}^{x_2} \frac{1}{2}(y_2 + y_1) dA = \frac{1}{2A}\int_{x_1}^{x_2} (y_2^2 - y_1^2) dx$$

假定 $y_2 = x + 2$ 和 $y_1 = x^2$，如图 5.7 所示。图中直观显示了交点为 $x_1 = -1$ 和 $x_2 = 2$。积分结果产生出 $A = 4.5, x_c = 0.5, y_c = 1.6$。再用数值方法重复这些计算。

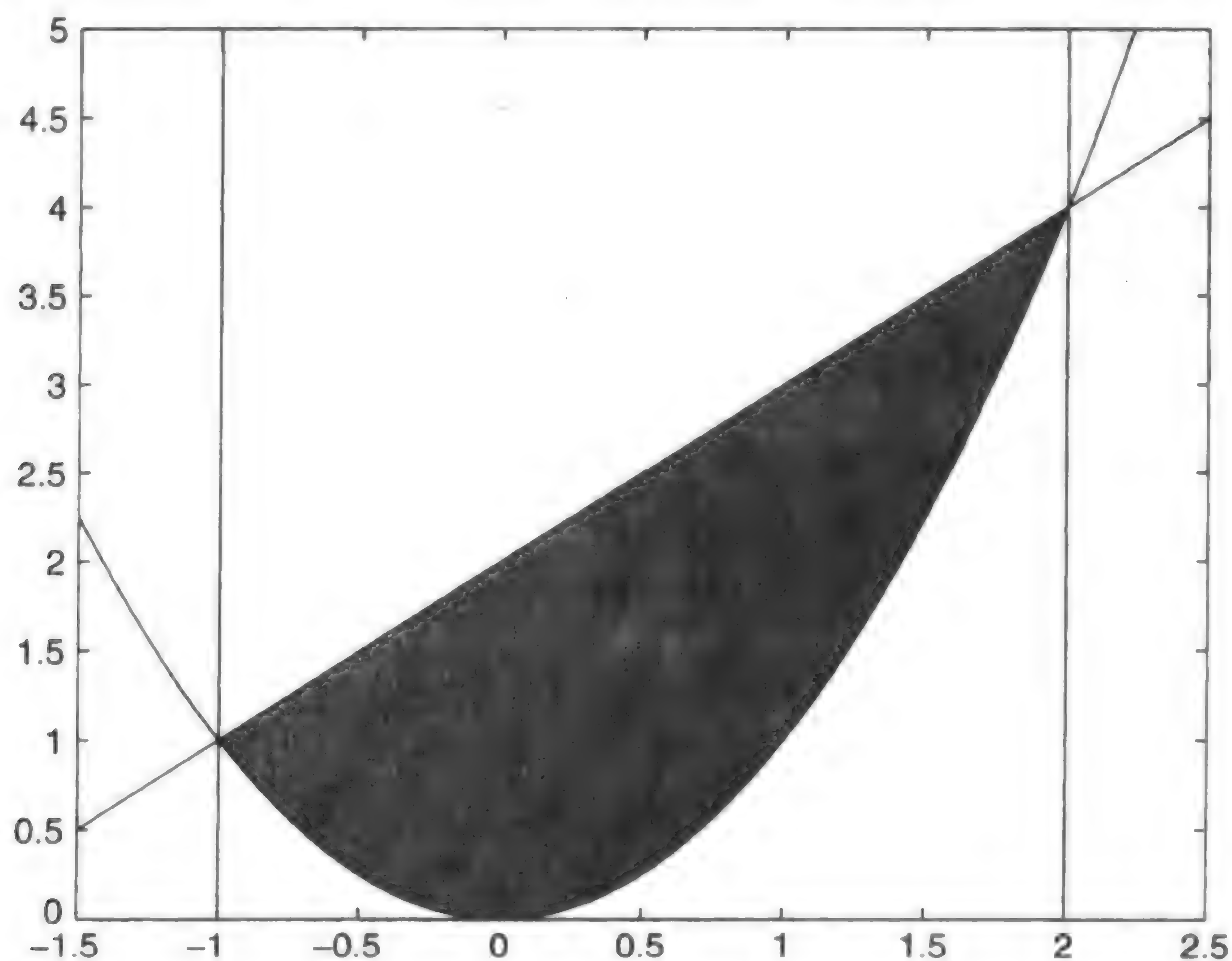


图 5.7　填充面积与重心的确定

因为这些曲线的表达式相对简单，所以用 inline 得到以下程序：

```
Atop = inline('x + 2','x');
Abot = inline('x.^2','x');
Area = quadl(Atop, - 1,2) - quadl(Abot, - 1,2)
Mxc = inline('x. * ((x + 2) - x.^2)','x');
Myc = inline('((x + 2).^2 - x.^4)/2','x');
xc = quadl(Mxc, - 1,2)/Area
yc = quadl(Myc, - 1,2)/Area
```

执行得到:

```
Area =
  4.5000
xc =
  0.5000
yc =
  1.6000
```

例 5.11 指数衰减正弦波曲线面积

阻尼正弦波用函数 **DampedSineWave** 表示。令 $\xi = 0.1$,在 $0 \leq \tau \leq 20$ 范围内求积分,程序如下:

```
xi = 0.1;
Area = quadl( @DampedSineWave,0,20,[ ],[ ],xi)
```

执行结果为:

```
Area =
  0.3022
```

所得结果与例 5.4 一致。

函数 dblquad 在容许误差 t_o 内,对用户提供的函数 $f(x,y)$ 在 x 轴方向从下限 x_1 到上限 x_u,在 y 轴方向从下限 y_1 到上限 y_u 进行数值积分。该函数也可以将 p_j 参数传递到定义 $f(x,y)$ 的函数中去。用函数文件表示 $f(x,y)$ 时,dblquad 的一般表达式为:

```
dblquad( @FunctionName, x1, xu, y1, yu, t0, meth, p1, p2, ... )
```

其中,**FunctionName** 是函数 M 文件不带后缀 .m 的文件名,x1 = x_1;xu = x_u;yl = y_1;yu = y_u;t0 = t_o(省略时,使用默认值);p1,p2,....等为 p_j 参数;当 meth = [ ]时,使用 quadl 函数,当函数由 inline 创建时,使用以下语句:

```
dblquad(InlineFunctionName, xi, xu, y1, yu, t0, meth, p1, p2, ... )
```

InlineFunctionName 是 inline 函数的名称。

函数文件的接口形式为:

```
function z = FunctionName(x,y,p1,p2,...)
Expression(s)
```

下面说明 dblquad 的使用。

例 5.12 两相关变量的概率

下面,对在给定区域内服从正态分布的两随机变量的概率表达式进行数值积分:

$$V = \frac{1}{2\pi\sqrt{1 - r^2}}\int_{-2}^{2}\int_{-3}^{3} e^{-(x^2 - 2rxy + y^2)/2}\,dx\,dy$$

假定 $r = 0.5$,程序为:

```
r=0.5;
Arg=inline('exp(-(x.^2-2*r*x.*y+y.^2))','x','y','r');
P=dblquad(Arg,-3,3,-2,2,[],[],r)/2/pi/sqrt(1-r^2)
```

执行程序得:

```
P=
  0.6570
```

5.5.3 常微分方程的数值求解函数——ode45

MATLAB 可以解决两种不同类型的常微分方程。函数的选择根据问题的类型而定,如果是初始值问题,条件 $t=0$(或 $x=0$)已知;如果是边界值问题,域的上下边界条件已知,即 $x=0$ 和 $x=L$。初始值问题由 ode45 函数实现求解,边界值问题由 bvp4c 函数求解。本节讨论 ode45 函数,在 5.5.4 节介绍 bvp4c 函数。

函数 ode45 返回 n 个一阶常微分方程的数值解:

$$\frac{dy_j}{dt} = f_j(t, y_1, y_2, \cdots, y_n) \qquad j = 1,2,\cdots,n$$

在 $t_0 \leqslant t \leqslant t_f$ 区间内,初始条件 $y_j(t_0) = a_j, j=1,2,\cdots,n$, a_j 为常数。ode45 的参数如下:

```
[t,y]=ode45(@FunctionName,[t0,tf],[a1,a2,…,an],options,p1,p2,…)
```

其中,输出 t 是一时间列向量,时间 t 满足 $t_0 \leqslant t \leqslant t_f$ 由 ode45 确定。输出 $\boldsymbol{y}$ 是数值解的矩阵,行对应时间 t,列对应解的值:

$$\begin{aligned} \boldsymbol{y}(:,1) &= y_1(t) \\ \boldsymbol{y}(:,2) &= y_2(t) \\ &\cdots \\ \boldsymbol{y}(:,n) &= y_n(t) \end{aligned}$$

ode45 的第一个参数 **@FunctionName** 是指向函数文件 *FunctionName.m* 的句柄,必须具有以下形式[①]:

```
function yprime=FunctionName(t,y,p1,p2,…)
```

其中,t 是自变量,$\boldsymbol{y}$ 是一向量,其元素对应于 y_j。$p1,p2,\cdots$ 是传递到 **FunctionName** 的参数,*yprime* 是一个 n 阶列向量,其元素是 $f_j(t, y_1, y_2, \cdots, y_n), j=1,2,\cdots,n$,记为:

$$yprime = [f_1; f_2; \cdots; f_n]$$

变量名 *yprime*,**FunctionName** 等由程序员指定。

ode45 的第二个参数是二元向量,给出获得数值解的开始和结束时间。这是一个时间向量 $[t_0 t_1 t_2 \cdots t_f]$,在其上得到解。第三个参数是一初始条件向量 $y_j(t_0) = a_j$。第四个参数 *options* 通常设定为空。然而,如果求解方法对应的某些误差要改变时,就要用到 odeset(参见 odeset 帮助文件)。其他参数都被传递到函数 **FunctionName**。

在 MATLAB 中还有 6 个已知初值的常微分方程求解命令。根据微分方程的特点,每个命令

---

① 也可用 inline 函数,但接口行的参数数量不同。这里不再讨论 inline 函数的用法,inline 接口与旧版 MATLAB 的 ode45 兼容以保证向后兼容性。

各有各的优点。这 6 个命令分别是 ode23,ode113,ode15s,ode23s,ode23t 和 ode23tb。其用法与 ode45 相同。详见 MATLAB 用户指南和各自的帮助文件。

通过求解单变量二阶常微分方程说明函数的用法:

$$\frac{d^2y}{dt^2} + 2\xi\frac{dy}{dt} + y = h(t) \tag{5.3}$$

初始条件 $y(0) = a, dy(0)/dt = b$。将上式重写为一个一阶方程组:

$$y_1 = y$$
$$y_2 = \frac{dy}{dt}$$

则系统的方程式为:

$$\frac{dy_1}{dt} = y_2$$
$$\frac{dy_2}{dt} = -2\xi y_2 - y_1 + h$$

初始条件 $y_1(0) = a, y_2(0) = b$。

考虑这种情况:$\xi = 0.15, y(0) = 1, dy(0)/dt = 0$ 且

$$h(t) = \sin(\pi t/5) \quad 0 \leqslant t \leqslant 5$$
$$= 0 \quad t > 5$$

只考虑 $0 \leqslant t \leqslant 35$ 范围内的解。有 $y_1(0) = 1, y_2(0) = 0, t_0 = 0, t_f = 35$。编写主函数 **Exampleode** 和子函数 **HalfSine** 实现求解。下面是求解常微分方程的程序(函数)及绘制的曲线 $y_1(t) = y(t)$:

```
function Exampleode
[t,yy] = ode45(@HalfSine,[0 35],[1 0],[],0.15);plot(t,yy(:,1))
function y = HalfSine(t,y,z)
h = sin(pi * t/5) .* (t <= 5);
y = [y(2); - 2 * z * y(2) - y(1) + h];
```

结果如图 5.8 所示:$yy(:,1) = y_1(t) = y(t), yy(:,2) = y_2(t) = dy/dt$。

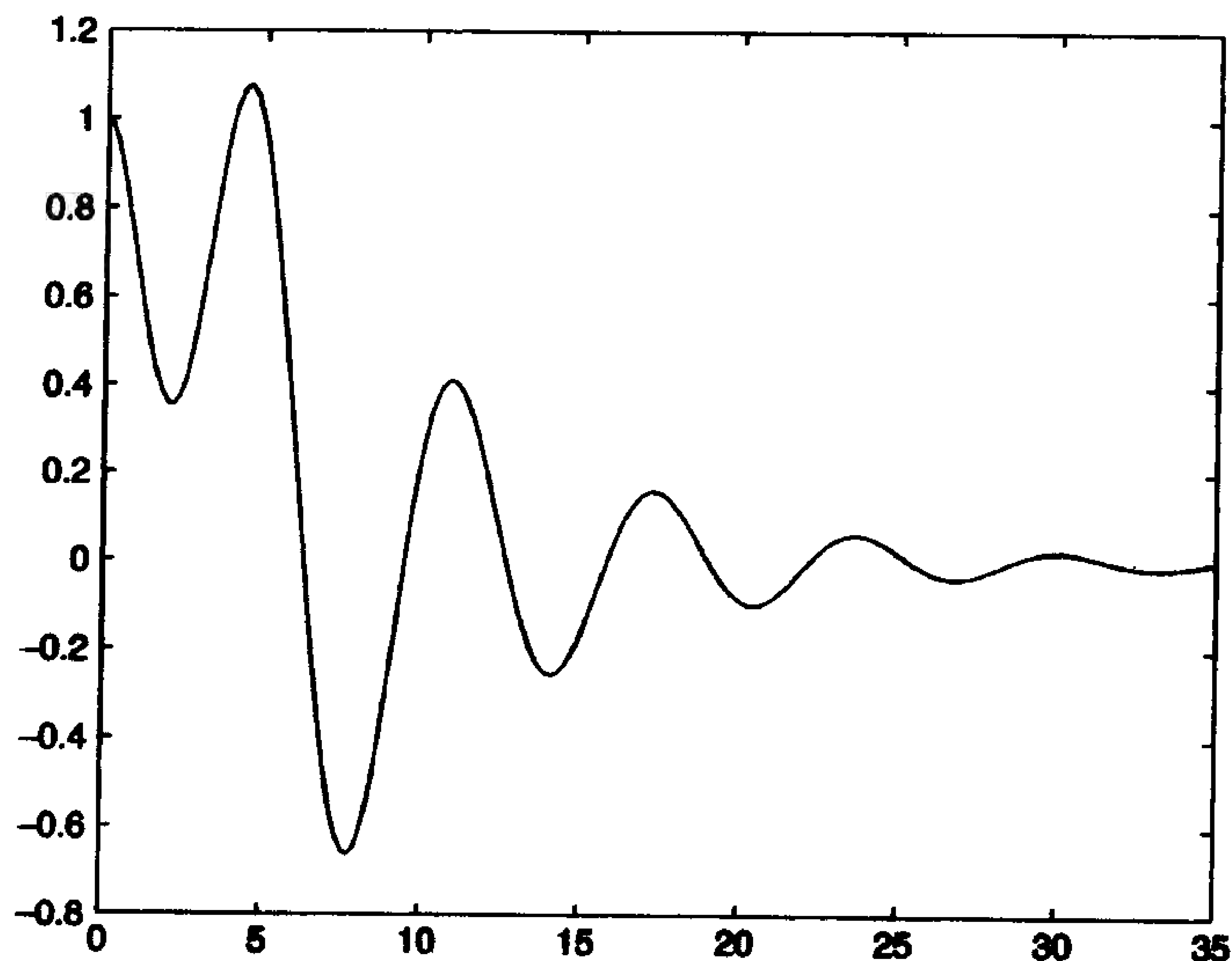


图 5.8 初始条件 $y(0) = 1$, $dy(0)/dt = 0$, 且 $h(t)$是半波正弦时,式(5.3)的响应曲线

举例说明 ode45 的扩展应用。

例5.13 竖直加热板的自然对流

下式用于说明当和冷液体接触时，竖直加热板的自然对流(参见12.3.2节)：

$$\frac{d^3f}{d\eta^3}+3f\frac{d^2f}{d\eta^2}-2\left(\frac{df}{d\eta}\right)^2+T^*=0$$

$$\frac{d^{2*}T^*}{d\eta^2}+3\Pr f\frac{dT^*}{d\eta}=0$$

当 $\Pr=0.7$，$\eta=0$ 时，边界条件为：

$$f=0 \qquad \frac{df}{d\eta}=0 \qquad \frac{d^2f}{d\eta^2}=0.68$$

$$T^*=1 \qquad \frac{dT^*}{d\eta}=-0.50$$

通过引入中间变量，系统可以分解为5个一阶方程构成的方程组：

$$y_1=f \qquad y_4=T^*$$

$$y_2=\frac{df}{d\eta} \qquad y_5=\frac{dT^*}{d\eta}$$

$$y_3=\frac{d^2f}{d\eta^2}$$

其中，y_1是流函数，y_2是速度，y_3是交量，y_4是温度，y_5是热流量。则有关这些新变量的微分方程是：

$$\frac{dy_1}{d\eta}=y_2 \qquad \frac{dy_4}{d\eta}=y_5$$

$$\frac{dy_2}{d\eta}=y_3 \qquad \frac{dy_5}{d\eta}=-3\Pr y_1y_5$$

$$\frac{dy_3}{d\eta}=2y_2^2-3y_1y_3-y_4$$

$\eta=0$ 处相应的边界条件为：

$$y_1(0)=0 \qquad y_4(0)=1$$

$$y_2(0)=0 \qquad y_5(0)=-0.50$$

$$y_3(0)=0.68$$

为解这一方程组，首先创建 **NaturalConv** 函数，指定代表5个一阶微分方程的等式右边的列向量。假定 $0\leq\eta\leq20$，程序为：

```
function Example 5_13 y0 = [0,0,0.68,1,-0.50];
Pr = 0.7;
[eta ff] = ode45(@NaturalConv,[0 20],y0,[],Pr);

function ff = NaturalConv(x,y,Pr)
ff = [y(2);y(3);-3*y(1)*y(3)+2*y(2)^2-y(4);y(5);-3*Pr*y(1)*y(5)];
```

结果如图12.14所示。

例5.14 钟摆阻尼器

如图5.9所示的钟摆阻尼器，系统的无量纲运动方程为①：

$$\ddot{z}+2\zeta_x\dot{z}+z+m_r[\ddot{\varphi}\sin\varphi+\dot{\varphi}^2\cos\varphi]=f_o\cos\Omega\tau$$

① 参见 B. Balachandran 和 E. B. Magrab, *Vibrations*, Brooks/Cole, Belmont, CA, 2004, 第458页。

$$\ddot{\varphi}+2\zeta_t\dot{\varphi}+(\omega_r^2+\ddot{z})\sin\varphi=0$$

点表示对 τ 的求导，g 为万有引力常数，c 和 c_t 为阻尼值，k 为弹性系数，m 和 M 分别为摆的质量和主质量，且有：

$$z=\frac{x}{l}\quad \tau=\omega_x t\quad \omega_r=\frac{\omega_\varphi}{\omega_x}\quad \Omega=\frac{\omega}{\omega_x}\quad f_o=\frac{F_o}{(M+m)l\omega_x^2}\quad \omega_\varphi=\sqrt{\frac{g}{l}}$$

$$2\zeta_x=\frac{c}{(M+m)\omega_x}\quad 2\zeta_i=\frac{c_t}{ml^2\omega_x}\quad m_r=\frac{m}{(M+m)}\quad \omega_x=\sqrt{\frac{k}{m+M}}$$

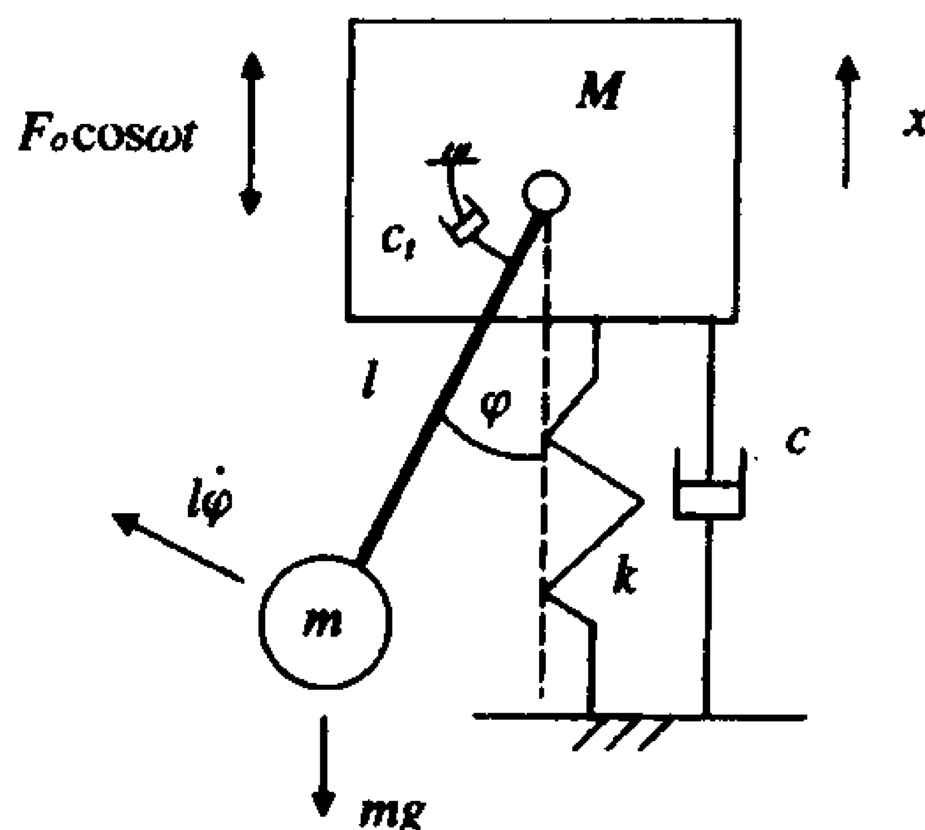


图 5.9 钟摆阻尼器

为将这些等式变为一阶方程式形式，设：

$$x_1=z\quad x_3=\varphi$$
$$x_2=\dot{z}\quad x_4=\dot{\varphi}$$

则有：

$$\dot{x}_1=x_2\quad \dot{x}_3=x_4$$

将上式代入原方程得到两个一阶方程式，用矩阵形式表示如下：

$$\begin{bmatrix}1 & m_r\sin x_3\\ \sin x_3 & 1\end{bmatrix}\begin{Bmatrix}\dot{x}_2\\ \dot{x}_4\end{Bmatrix}=\begin{Bmatrix}A\\ B\end{Bmatrix}$$

其中，

$$\begin{Bmatrix}A\\ B\end{Bmatrix}=\begin{Bmatrix}f_o\cos\Omega\tau-2\zeta_x x_2-x_1-x_4^2 m_r\cos x_3\\ -2\zeta_1 x_4-\omega_r^2\sin x_3\end{Bmatrix}$$

然后求 $\dot{x}_2$和 $\dot{x}_4$：

$$\dot{x}_2=\frac{A-Bm_r\sin x_3}{1-m_r\sin^2 x_3}$$

$$\dot{x}_4=\frac{B-A\sin x_3}{1-m_r\sin^2 x_3}$$

实际情况下，m_r 远小于 1，因此不会出现异常。

下面对方程求解，条件为 $\Omega=1,\omega_r=0.5,\zeta_x=0.05,\zeta_t=0.005,m_r=0.05,f_o=0.03$ 及 $0\leqslant\tau\leqslant300$。同时，画出 $\varphi(0)=x_3=0.02$ rad 时的角位移 φ 的曲线。由于函数所需参数有 6 个，因此使用 global 定义全局变量。另外，ode45 函数接口需要的子函数为 **PendulumAbsorber**。程序为：

```
function Example5_14
global MR WR ZX ZT OM FO
```

```
MR = 0.05;ZX = 0.05;ZT = 0.005;
OM = 1;WR = 0.5;FO = 0.03;
[t w] = ode45(@PendulumAbsorber,[0 300],[0 0.02 0]);
plot(t,w(:,3))

function Q = PendulumAbsorber(t,w)
global MR WR ZX ZT OM FO
A = FO * cos(OM * t) - w(1) - 2 * ZX * w(2) - MR * w(4)^2 * cos(w(3));
B= -2 * ZT * w(4) - WR^2 * sin(w(3))^2);
x4dot = (B - A * sin(w(3)))/(1 - MR * sin(w(3))^2);
x2dot = (A - MR * B * sin(w(3)))/(1 - MR * sin(w(3))^2);
Q = [w(2):x2dot;w(4);x4dot];
```

结果绘成的曲线如图 5.10 所示。

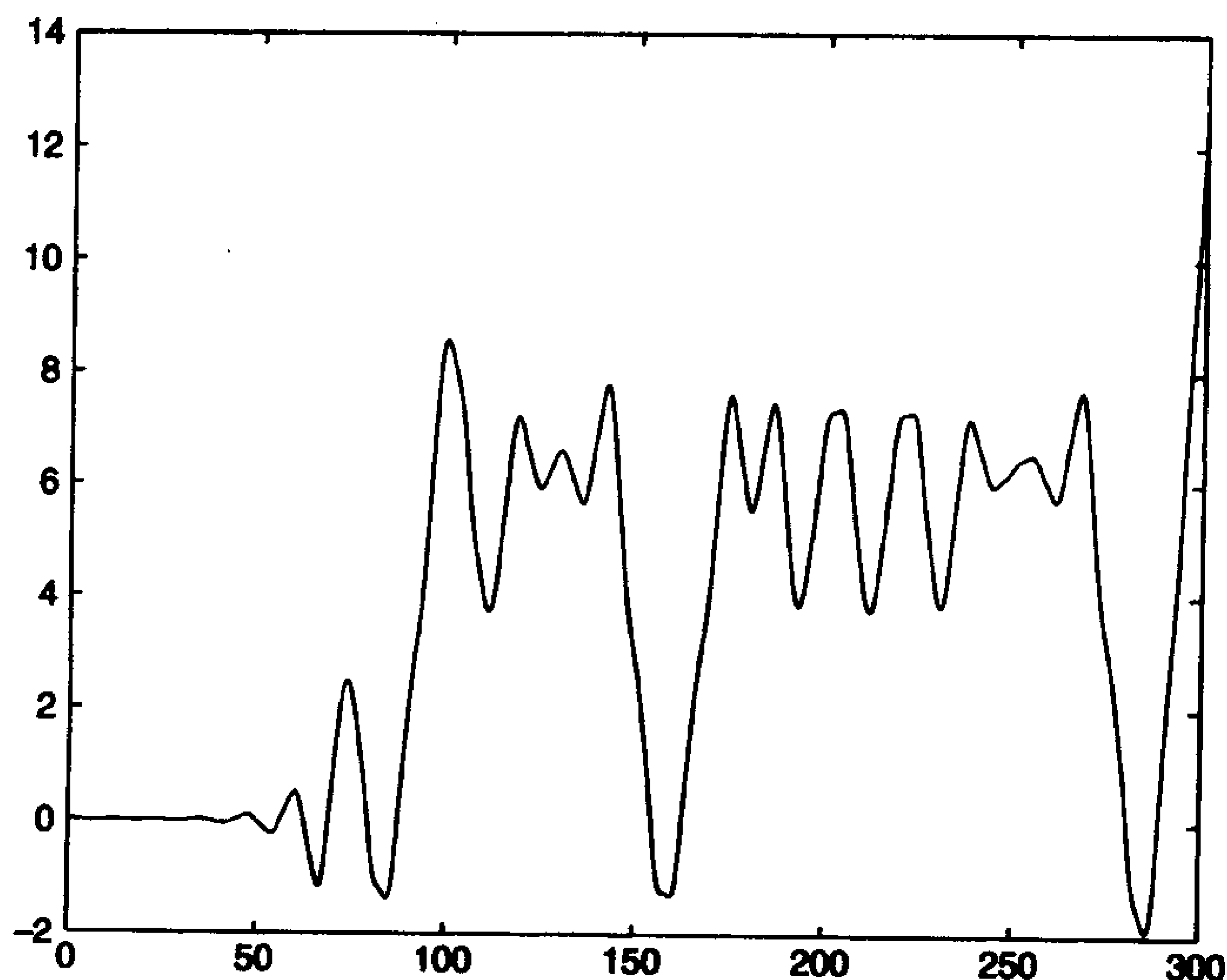


图 5.10　钟摆阻尼器的摆动角位移轨迹

5.5.4　常微分方程的数值求解函数——bvp4c

bvp4c 针对两点边界值问题进行数值求解。但与 ode45 不同，使用 bvp4c 函数时需要同时使用其他几个 MATLAB 函数，如对 bvp4c 进行初始化的函数 bvpinit 及使输出图形更光滑的函数 deval。除此之外，还需要创建其他几个函数。

bvp4c 函数实现对以下 n 个一阶常微分方程的数值求解：

$$\frac{dy_j}{dx} = f_j(x, y_1, y_2, \cdots, y_n, q) \quad j = 1,2,\cdots,n$$

x 的取值范围为 $a \leqslant x \leqslant b$，边界条件为 $y_j(a) = a_j$ 及 $y_j(b) = b_j$，其中 $j = 1,2,\cdots,n$，a_j 和 b_j 为常数，q 为未知参数向量。这里将不讨论方程式含有未知参数 q 的情况。bvp4c 的参数和输出如下：

```
sol = bvp4c(@FunctionName, @BCFunction, solinit, options, p1, p2, ...)
```

FunctionName 函数的接口如下：

```
function dydx = FunctionName(x,y,p1,p2,...)
```

其中,x 为一标量,y 为f_j 列向量,p1, p2 等为 f_j 的参数,参数值已知,输出 ***dxdy*** 是列向量。

BCFunction 函数包含边界条件 $y_j(a)=a_j$ 及$y_j(b)=b_j, j=1,2,\cdots,n$。其接口为:

```
function Res = BCFunction(ya,yb,p1,p2,...)
```

ya 是表示$y_j(a)$的列向量,***yb*** 为表示$y_j(b)$的列向量。即使不要求边界条件,已知参数 p1, p2 等也必须出现在接口定义语句中。输出 ***Res*** 为列向量。

变量 *solinit* 为一结构[①],可以由函数 bvpinit 得到:

```
solinit = bvpinit(x,y)
```

x 向量为初始网格点的估计值,向量 y 为每个y_j 的估计值。向量 x 和y 的长度互不相关。

bvp4c 的输出 *sol* 是一结构,为特定数量点对应的解。为使曲线变得更光滑,需要在中间插入一些点,使用:

```
sxint = deval(sol,xint)
```

xint 为点向量,函数 deval 根据这些点向量求解。*sol* 为函数 bvp4c 的输出。

为说明 bvp4c 的使用,考虑常微分方程:

$$\frac{d^2y}{dx^2}+ky=x \qquad 0\leqslant x\leqslant 1$$

边界条件为:

$$y(0)=0$$
$$y(1)=0$$

首先,将方程转换成两个一阶微分方程,引入中间变量:

$$y_1=y$$
$$y_2=\frac{dy}{dt}$$

得:

$$\frac{dy_1}{dx}=y_2$$
$$\frac{dy_2}{dx}=x-ky_1$$

边界条件为:

$$y_1(0)=0$$
$$y_1(1)=0$$

下面创建函数和子函数。实现一阶常微分方程的子函数为 **OdeBvp**, 记录边界条件的函数为 **OdeBC**。初始估计值由 bvpinit 给出。在 0 和 1 之间选择 5 个点,假定 y_1 有一个解为 -0.05, y_2 有一个解为 0.1,设 $k=2$,则程序为:

```
function bvpExample
k = 2;
solinit = bvpinit(linspace(0,1,5),[ -0.05,0.1]);
exmpsol = bvp4c(@OdeBvp,@OdeBC,solinit,[],k);
x = linspace(0,1,50);
y = deval(exmpsol,x);
plot(x,y(1,:))
```

---

① 随后将举例说明结构。

```
function dydx = OdeBvp(x,y,k)
dydx = [y(2);x - k * y(1)];

function res = OdeBC(ya,yb,k)
res = [ya(1);yb(1)];
```

根据结果绘成的图形如图 5.11 所示。

现在，讨论程序中各个函数的输出。*exmpsol* 是一结构，通过 *exmpsol* 结构可以访问 *exmpsol*.$y(1,:)$和 *exmpsol*.$y(2,:)$。*exmpsol*.$y(1,:)$给出由 *exmpsol*.x 确定的网格点的 y_1 值，*exmpsol*.$y(2,:)$给出相同点的 y_2 值。执行计算过程时，bvp4c 产生这些量。该例中 bvp4c 使用的网格点个数为 7，此外，deval 的输出 y 更类似于 ode45 的输出。变量 y 为一(2×50)数组，$y(1,:)=y_1=y$，$y(2,:)=y_2=dx/dy$。由结构 *exmpsol*.$y(1,:)$的 7 个值所绘制的曲线如图 5.11 所示。

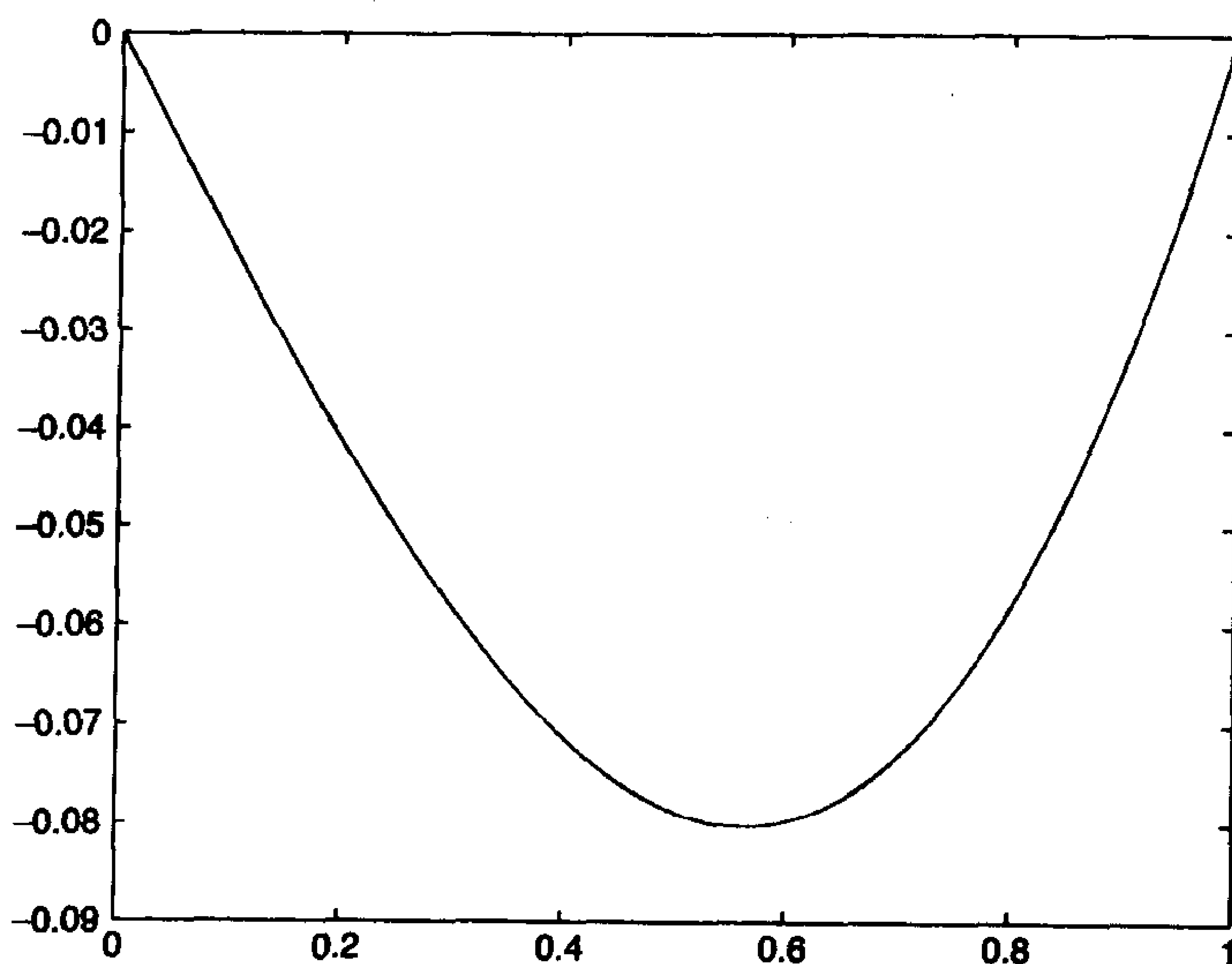


图 5.11　例 5.15 的 $y(x)$曲线

在例 5.15 中，以一个两端固定均匀受力的梁的静态位移计算为例，进一步阐述 bvp4c 函数的使用①。

例 5.15　质量分布均匀的梁的位移

根据 8.2.1 节，受力于静止负荷 q(无量纲)的梁的位移 y(无量纲)的控制方程为：

$$\frac{d^4 y}{d\eta^4}=q(\eta)$$

其中，$0\leqslant\eta\leqslant1$，且：

$$\theta=\frac{dy}{d\eta}$$

$$M=\frac{d^2 y}{d\eta^2}$$

① 要参考其他例子，参见 L. F. Shampine, M. W. Reichelf 和 J. Kieraenka, "Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB With bvp4c"，可通过以下网址得到：ftp://ftp.mathworks.com/pub/doc/papers/bvp/。

$$V=\frac{d^3y}{d\eta^3}$$

分别为无量纲斜率、力矩及切力。

引入以下关系式,将上述方程转换为一系列一阶常微分方程:

$$y_1=y \quad y_3=\frac{d^2y}{d\eta^2}$$

$$y_2=\frac{dy}{d\eta} \quad y_4=\frac{d^3y}{d\eta^3}$$

综合这些关系式和初始常微分方程,得到:

$$\frac{dy_1}{d\eta}=y_2 \quad \frac{dy_3}{d\eta}=y_4$$

$$\frac{dy_2}{d\eta}=y_3 \quad \frac{dy_4}{d\eta}=q$$

考虑每端受力的梁的情况,即每端的位移和力矩均为 0,有 $y(0)=y(1)=d^2y(0)/d\eta^2=d^2y(1)/d\eta^2=0$。此外,在 $\eta=1$ 端施加一额外力矩 M_r,沿梁长施加均匀负荷 q_o,有 $q(\eta)=q_o$,在 $\eta=1$ 处的力矩边界条件为 $d^2y(1)/d\eta^2=M_r$。

假定 $M_r=0.8,q_o=1.0$。估计 10 个网格点,且位移、斜率、力矩及切力的大小均为 0.5。此外,表示系统一阶方程的函数为 **BeamODEqo**,表示边界条件的函数为 **BeamHingedBC**。程序为:

```
function Example5_15
qo = 1;Mr = 0.8;
solinit = bvpinit(linspace(0,1,10),[0.5,0.5,0.5,0.5]);
beamsol = bvp4c(@BeamODEqo,@BeamHingedBC,solinit,[],qo,Mr);
eta = linspace(0,1,50);
y = deval(beamsol,eta);
plot(eta,y(1,:))

function dydx = BeamODEqo(x,y,qo,Mr)
dydx = [y(2);y(3);y(4);qo];

function bc = BeamHingedBC(y0,y1,qo,Mr)
bc = [y0(1);y0(3);y1(1);y1(3) - Mr];
```

结果如图 5.12 所示。

8.2.1 节将这些结果推广到了组合不同边界条件及不同负荷类型情况下的应用。

5.5.5 求局部极小值的函数——fminbnd

利用 fminbnd 函数可以求出 $f(x)$在区间 $a\leqslant x\leqslant b$ 内误差小于 t_o的极小值,并且能将参数 p_j传递到定义$f(x)$的函数中去。fminbnd 的一般表达式为:

```
[xmin fmin] = fminbnd(@FunctionName,a,b,options,p1,p2,...)
```

其中,**FunctionName** 是函数文件名(没有后缀“.m”),a = a,b = b,***options*** 是一可选择向量,其参数用 optimset 设置(参见 optimset 的帮助文件)。此外,p1,p2 等都称为参数 p_j。*xmin* 是 **FunctionName** 表示的函数取最小值时的 x 值,*fmin* 为在 *xmin* 处的 **FunctionName** 的值。

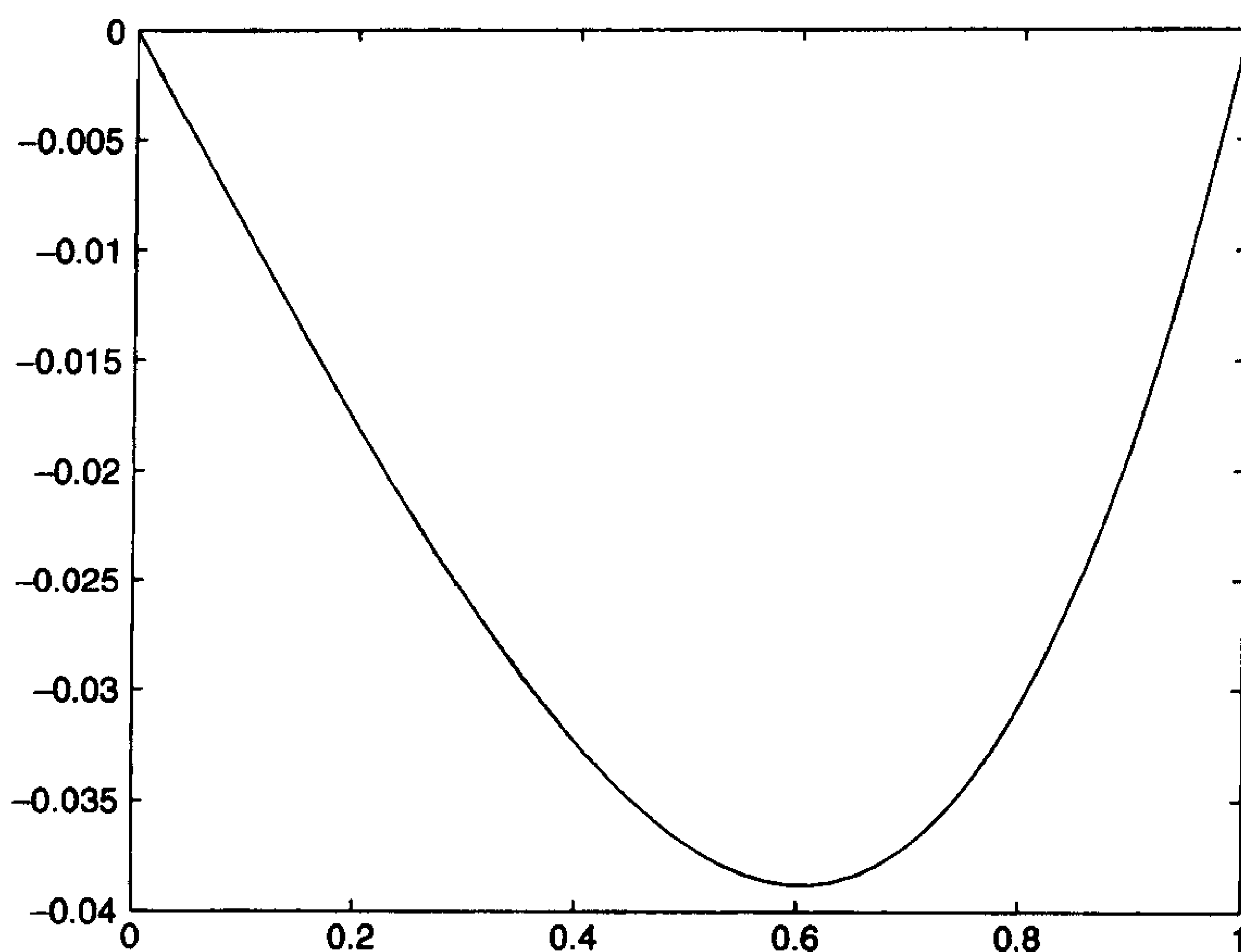


图 5.12 均匀分布受力梁在 $\eta=1$ 处施加额外力矩时的位移曲线

该函数的接口形式如下：

```
function z = FunctionName(x,p1,p2,...)
Expression(s)
```

其中，x 是自变量，通过改变 x 值 fminbnd 不断使 $f(x)$ 极小化。自变量必须出现在这个位置。现在说明 fminbnd 的用法。

$f(x)$ 表达式由 inline 函数表示，其名称为 **InlineFunctionName**，fminbnd 语句如下：

```
[xmin,fmin] = fminbnd(InlineFunctionName,a,b,options,p1,p2,...)
```

如图 5.13 所示，MATLAB 示例函数 humps 在 $0.5\leqslant x\leqslant0.8$ 之间的极小值由下面的语句求出：

```
options = optimset('Display','off');
[xmin,fmin] = fminbnd(@humps,0.5,0.8,options)
```

执行程序得到如下结果：

```
xmin =
  0.6370
fmin =
  11.2528
```

因此，在区间 $0.5\leqslant x\leqslant0.8$ 上的 $x=0.6370$ 处，函数存在极小值 11.253。

另一方面，如果要找到函数 humps 在区间 $0\leqslant x\leqslant0.5$ 上的极大值及 x 的值，则用 fminbnd 进行负值操作，或求 humps 的倒数。用 inline 函数创建函数计算 humps 在该区域内的负值。程序如下：

```
options = optimset('Display','off');
[xmax,fmax] = fminbnd(inline('-humps(x)','x'),0,0.5,options);
disp(['Maximum value of humps in the interval 0 <= x <= 0.5 is ' num2str (-fmax)])
disp(['which occurs at x = ' num2str (xmax)])
```

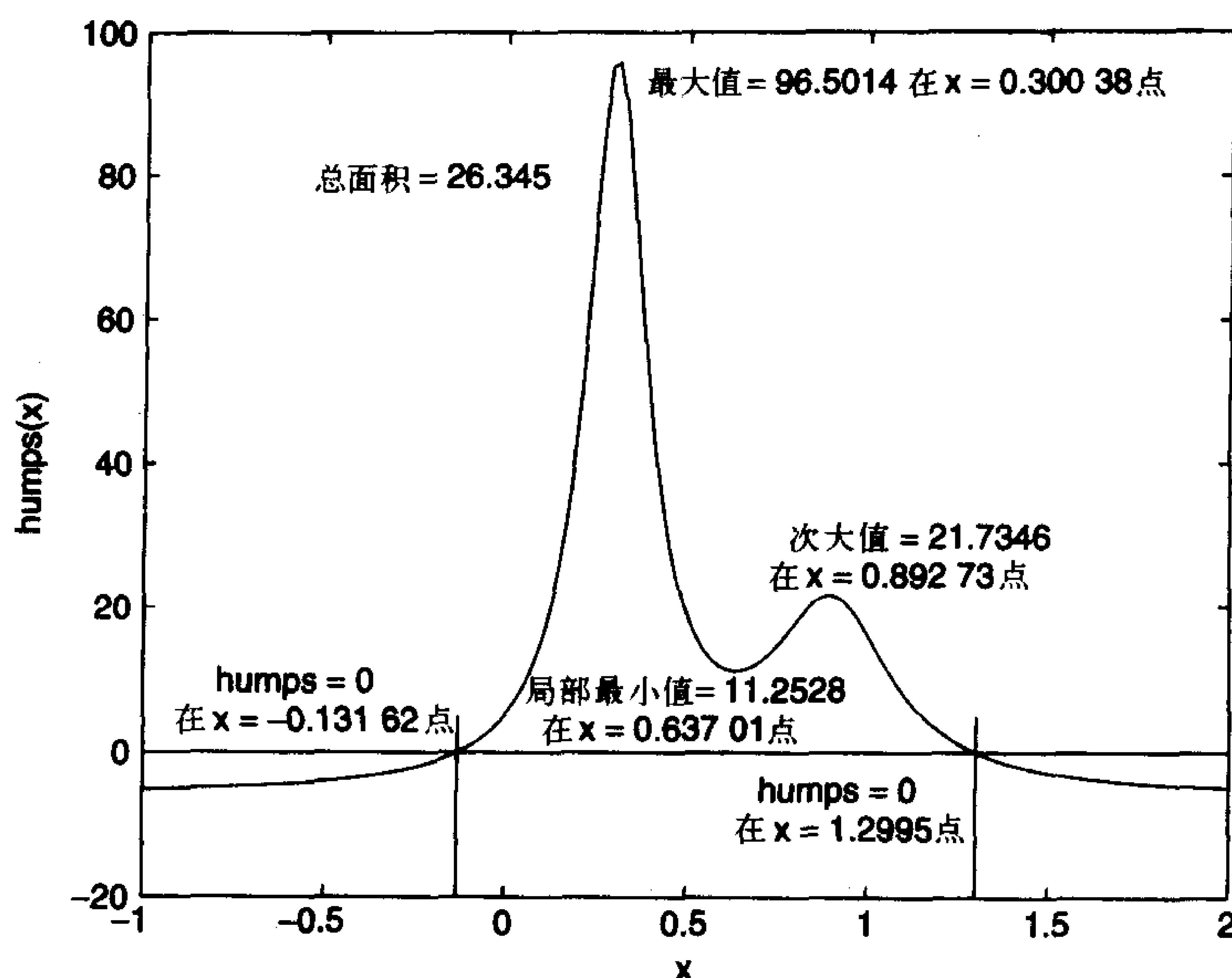


图 5.13　MATLAB 示例函数 humps 的属性

在命令行窗口上显示的执行结果为:

```
Maximum value of humps in the interval 0 <= x <= 0.5 is 96.5014
which occurs at x = 0.30039
```

fminbnd 使用 humps 的负函数计算,所以显示前必须先计算 *fmax* 的负值。利用 5.5.1 节和 5.5.2 节讨论的方法,可验证图 5.13 中的其他量。

5.5.6　非线性方程的数值求解函数——fsolve

优化工具箱中的内置函数 fsolve 用于求解非线性方程,如 n 元非线性方程组 $f_n(x_1, x_2, \cdots, x_n) = 0$, x_n 的估计值为 $x_s = [x_{s1}\ x_{s2} \cdots x_{sn}]$。函数输出为有 n 个值的根向量 x_{roots}。该函数也可以将参数 p_j 传递到定义 $f_n(x)$ 的函数中。fsolve 的一般表示为:

```
xroots = fsolve(@FunctionName, xs, options, p1, p2, ...)
```

其中,**FunctionName** 是函数名(没有后缀".m"), xs = x_s, *options* 是可选择的向量,其参数由 optimset 设置(参见帮助文件中的 optimset)。$p1, p2$ 等统称为参数 p_j。函数的输出 *xroots* 为 x_{roots} 向量。

该函数的接口形式如下:

```
function z = FunctionName(x, p1, p2, ...)
z = [f1; f2; ...; fn];
```

其中, x 是由 n 个量 x_n 组成的向量, z 是一列向量,由 x 和参数 p_j 的 n 个非线性方程 $f_n(x_1, x_2, \cdots, x_n)$ 形式的 MATLAB 表达式组成。

如果 $f_n(x_1, x_2, \cdots, x_n)$ 由 inline 表示,其名称为 **InlineFunctionName**, fsolve 的访问形式为:

```
xroots = fsolve(InlineFunctionName, a, b, options, p1, p2, ...)
```

下面介绍 fsolve 的使用。

例5.16　倒置运动

考虑以下方程组，是图2.8所示的三自由度连杆机构倒置运动的中间求解过程，

$$r_1 - a_1\cos(\theta_1) - a_2\cos(\theta_1 + \theta_2) = 0$$

$$r_2 - a_1\sin(\theta_1) - a_2\sin(\theta_1 + \theta_2) = 0$$

为求解这一方程组，首先创建一函数 **kinematics**，将这些等式变为 fsolve 所要求的形式，如下所示：

```
function w = kinematics(theta,a1,a2,r1,r2)
w = [a1 * cos(theta(1)) + a2 * cos(theta(1) + theta(2)) - r1;...
    a1 * sin(theta(1)) + a2 * sin(theta(1) + theta(2)) - r2];
```

其中，$theta(1) = \theta_1$，$theta(2) = \theta_2$。

令 $r_1 = 1.8, r_2 = 2.1, a_1 = 1.0, a_2 = 2$ 且 θ_1和 θ_2的初值设为 $\pi/6$，则程序如下：

```
options = optimset('display','off');
z = fsolve(@kinematics,[pi/6 pi/6],options,1,2,1.8,2.1) * 180/pi;
for k = 1:length(z)
  disp(['Theta(' num2str(k,1) ') = ' num2str(z(k))' degrees'])
end
```

执行后给出：

```
Theta(1) = 16.6028 degrees
Theta(2) = 48.5092 degrees
```

因此，$\theta_1 = z(1) = 16.6028°$，$\theta_2 = z(2) = 48.5092°$。当 θ_1, θ_2的初值设为 π 时，可求出另一组值。因而使用 fsolve 时必须加以注意，特别是存在多个解时。

例5.17　抛物线与椭圆相交

考虑一椭圆与一条抛物线相交的情形。椭圆的表达式如下：

$$g(x,y) = x^2/4 + y^2 - 1 - 0$$

抛物线的表达式为：

$$f(x,y) = y - 4x^2 + 3 - 0$$

这两个函数的图形存在四个交点，因而 fsolve 的返回值与初始假设值有关系。fsolve 使用的函数由 inline 创建，其中 $xy(1) = x, xy(2) = y$。设初始值 $x = 0.5, y = -0.5$，求解的程序如下：

```
fgsolve = inline('[0.25 * xy(1).^2 + xy(2).^2 - 1;xy(2) - 4 * xy(1).^2 + 3]','xy');
options = optimset('Display','off');
xy = fsolve(fgsolve,[0.5, -0.5],options)
```

执行后得到：

```
xy =
  0.7188   -0.9332
```

因此，$x = xy(1) = 0.7188, y = xy(2) = -0.9332$。如果选择初值 $x = -0.5, y = 0.5$，则得到 $x = xy(1) = -0.9837, y = xy(2) = 0.8707$。

5.6 符号工具箱与函数的创建

符号运算工具箱提供了执行代数、矩阵、微积分符号操作时对符号的处理能力。将符号操作结果与 MATLAB 创建函数的能力结合起来,会获得一种对符号表达式进行数值计算的非常有效的方法。本节介绍符号运算工具箱提供的处理基本操作的函数,然后论述怎样直接将结果转换为数值。

下面将举例说明符号工具箱语法、符号表达式转换为函数、符号微分与积分、符号极限及反拉普拉斯变换的符号运算。并举例说明对这些结果的使用。

语法

创建符号变量的快捷方式是使用

```
syms a b c
```

该语句定义的 a, b 和 c 为符号变量。如要限制变量为实变量,则将语句修改为:

```
syms a b c real
```

这些符号可以与无符号变量、数字、MATLAB 函数及符号表达式混合使用。

考虑以下关系式:

$$f = 11.92e^{-a^2} + b/d$$

假定 $d = 4.2$,以符号形式表示此表达式的程序为:

```
syms a b
d=4.2;
f=11.92*exp(-a^2)+b/d
```

执行时显示:

```
f=
298/25*exp(-a^2)+5/21*b
```

其中,f 是一个符号对象。注意,21/5 = 4.2 及 298/25 = 11.92。符号表达式中的数值常转换为两个整数的比值。如果希望用十进制小数形式表示,使用:

```
vpa(f, d)
```

f 是符号表达式,d 是数字的有效位数。用 5 位有效数字表示十进制小数的程序为:

```
syms a b
d=4.2;
f=vpa(11.92*exp(-a^2)+b/d,5)
```

执行程序产生:

```
f=
11.920*exp(-1.*a^2)+.23810*b
```

将符号表达式转换为函数

如要将符号表达式 f 转换为函数,则使用 inline 和

```
vectorize(f)
```

vectorize 将参数转换为字符串,将乘、除、指数运算转换为对应的点操作。修改后的程序如下:

```
syms a b
d=4.2;
f=vpa(11.92*exp(-a^2)+b/d,5);
g=inline(vectorize(f),'a','b')
```

执行结果显示:

```
g=
  Inline function:
  g(a,b)=11.920.*exp(-1.*a.^2)+.23810.*b
```

换句话说,符号表达式 f 转换为函数 $g(a,b)$ 后,就可以像使用其他函数一样使用 $g(a,b)$。例如 $a=1$, $b=2$ 时,程序为:

```
syms a b
d=4.2;
f=vpa(11.92*exp(-a^2)+b/d,5);
g=inline(vectorize(f),'a','b');
z=g(1,2)
```

在命令窗口中显示:

```
z =
  4.8613
```

微分和积分

实现求微分的函数为:

```
diff(f,x,n)
```

$f=f(x)$是符号表达式,x 为一变量,在其上求微分,n 为求微阶次。例如,$n=2$ 时是求 2 阶微分。

下面以求 $b\cos(bt)$的微分为例说明函数的使用。先对 t 执行求导计算,再对 b 执行求导计算。程序为:

```
syms b t
dt=diff(b*cos(b*t),t,1)
db=diff(b*cos(b*t),b,1)
```

执行程序得到:

```
dt=
-b^2*sin(b*t)
dt=
cos(b*t)-b*sin(b*t)*t
```

实现求积分的函数为:

```
int(f, x, c, d)
```

$f=f(x)$是符号表达式,x 为一变量,在其上求积分,c 是积分的下限,d 为上限。省略 c 和 d 时,int 函数求 $f(x)$的不定积分。

通过对上述程序获得的微分结果再进行积分来说明 int 的使用。程序为:

```
syms b t
f=b*cos(b*t);
dt=diff(f,t,1);
db=diff(f,b,1);
it=int(dt,t)
ib=int(db,b)
```

程序执行结果为:

```
it=
b*cos(b*t)
ib=
1/t*sin(b*t)-1/t*(sin(b*t)-b*t*cos(b*t))
```

由此看到,*it* 就是对 *t* 进行求微分之前的初始函数 *f*。第二个函数与对 *b* 求微分前的 *f* 函数不太一致,但如果使用 simplify 函数进行化简,则会得到正确结果。修改后的程序如下:

```
syms b t
f=b*cos(b*t);
dt=diff(f,t,1);
db=diff(f,b,1);
it=int(dt,t);
ib=int(db,b);
ib=simplify(ib)
```

执行时得到:

```
ib=
b*cos(b*t)
```

在使用化简函数之前,通常要对其应用顺序及选择哪一个函数做出测试。关于一系列化简函数及其语法可以在帮助文件附录 contents 的"Function Reference in Symbolic Math Toolbox"中找到。

极限运算与代换

当独立变量趋向于一个特定值时,可以求符号表达式的极限。求极限的函数为:

```
limit(f,x,z)
```

$f=f(x)$是符号函数,对其执行求极限运算,x 是值趋近于 z 的符号变量。

为说明函数的使用方法,考虑以下表达式:

$$\lim_{a \to \infty}\left(\frac{2a+b}{3a-4}\right)$$

程序为:

```
syms a b
Lim=limit((2*a+b)/(3*a-4),a,inf)
```

inf 表示无穷(参见表 1.4)。执行程序得到:

```
Lim =
2/3
```

举另一个例子。考虑求极限公式：

$$\lim_{x\to 0}\left(\frac{\sin ax}{x}\right)$$

实现对上式求极限的程序为：

```
syms a x
Lim = limit(sin(a * x)/x,x,0)
```

执行时得：

```
Lim =
a
```

Limit 还可以用来执行符号表达式变量的代换操作。对于前例，这次要实现代换操作，$x=c+2$，程序为：

```
syms a x c
Lim = limit(sin(a * x)/x,x,c + 2)
```

执行时得：

```
Lim =
sin(a * (c+2))/(c+2)
```

在 9.4.1 节使用了 limit 函数。

反拉普拉斯变换

实现反拉普拉斯变换的函数为：

```
ilaplace(F,s,t)
```

其中，$F=F(s)$，s 为拉普拉斯变换参数，t 为反变换空间的变量名，通常代表时间。

为说明 ilaplace 函数如何使用，考虑式(5.3)。在 $y(0)=0$ 及 $dy(0)/dt=0$ 时，其拉普拉斯变换为：

$$Y(s)=\frac{H(s)}{s^2+2\xi s+1}$$

$Y(s)$是 $y(t)$的拉普拉斯变换，$H(s)$是 $h(t)$的拉普拉斯变换，ξ 为一实常数，$\xi<1$。如果 $h(t)=u(t)$，$u(t)$为单位阶跃函数，则 $H(s) = 1/s$。求 $y(t)$的程序为：

```
syms s t
syms xi real
den = s * (s^2 + 2 * xi * s + 1);
yt = ilaplace(1/den,s,t)
```

执行程序得：

```
yt =
1 + 1/(4 * xi^2 - 4)^(1/2) * (1/( - xi + 1/2 * (4 * xi^2 - 4)^(1/2)) * exp(( - xi + 1/2 * (4 * xi^2 - 4)^
(1/2)) * t) - 1/( - xi - 1/2 * (4 * xi^2 - 4)^(1/2)) * exp(( - xi - 1/2 * (4 * xi^2 - 4)^(1/2)) * t))
```

现在有两种方法可供选择：一种方法是先将 yt 转换为函数，然后将结果绘成图形；另一种方法是进一步化简 yt 以得到更精确的数学表达式。下面，将对这两种方法分别加以介绍。注意，$\xi<1$ 时，有：

$$(4*xi\hat{}2-4)\hat{}(1/2))\rightarrow\sqrt{4\xi^2-4}=2j\sqrt{1-\xi^2}$$

且 $\xi<1$。

将 yt 转换为函数并绘图,其程序为:

```
syms s t
syms xi real
den = s * (s^2 + 2 * xi * s + 1);
yt = ilaplace(1/den,s,t);
yoft = inline(vectorize(yt),'t','xi');
t = linspace(0,20,200);
plot(t,real(yoft(t,0.15)))
```

其中,假设 $\xi = 0.15$。对数值结果的检验表明,解的虚部实际上为 0。因此,使用 real 移去由舍入误差引起的残余虚部。结果如图 5.14 所示。

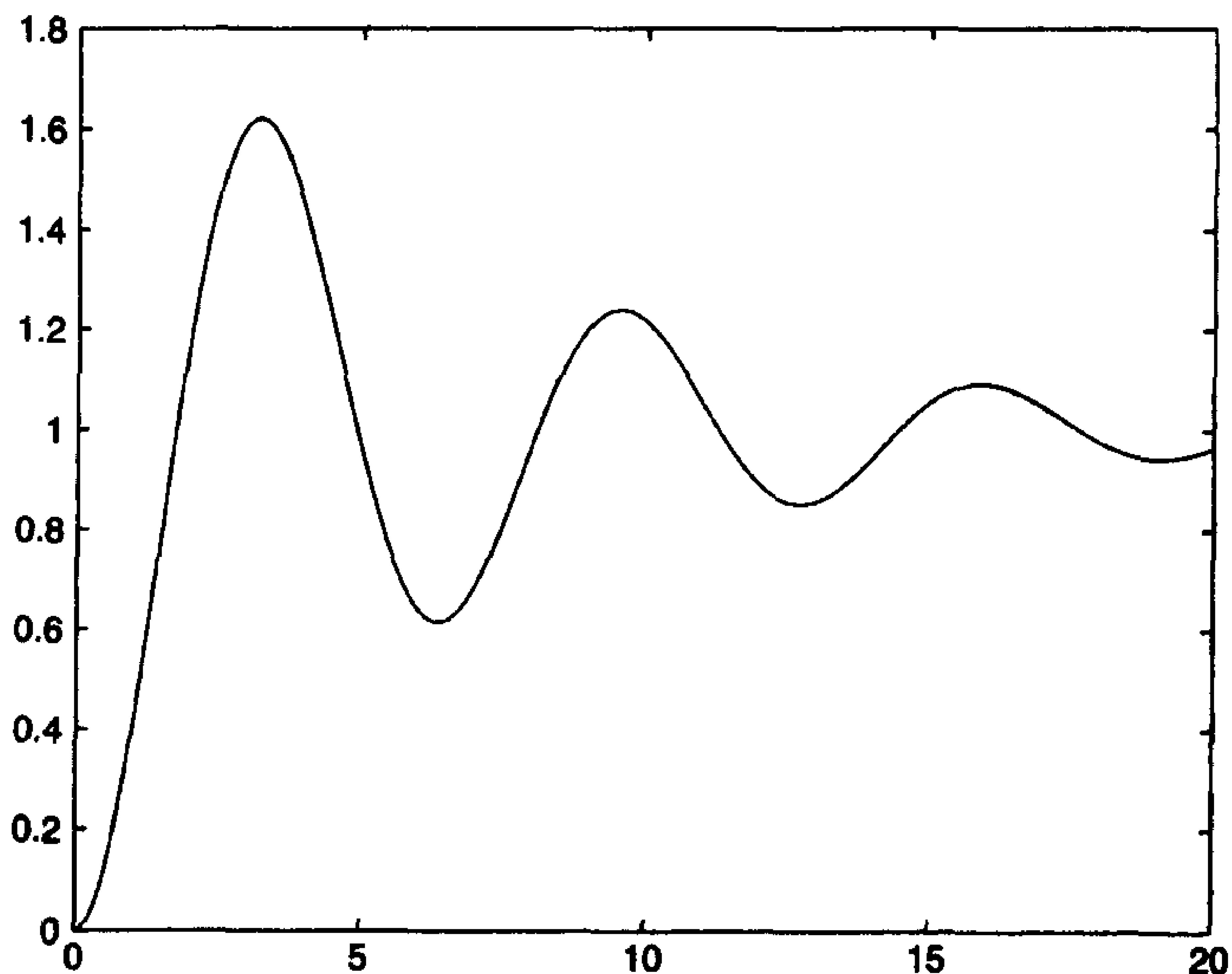


图 5.14 当 $h(t)=u(t)$, $\xi=0.15$ 时,使用拉普拉斯变换求解式(5.3)的结果

进一步化简 yt。首先,令:

$$(4*xi\hat{}2-4)\hat{}(1/2))\rightarrow 2jr$$

其中,

$$r=\sqrt{1-\xi^2}$$

用 subs 实现这一代换。subs 可以将每次出现的(4 * xi^2 - 4)^(1/2)用 2 * j * r 进行符号代换。使用

```
simple
```

函数得到最简结果。

程序修改为:

```
syms s t
syms xi r real
den = s * (s^2 + 2 * xi * s + 1);
yt = ilaplace(1/den,s,t);
```

```
yt1 = subs(yt,'(4 * xi^2 - 4)^(1/2)','2 * j * r');
yt2 = simple(yt1)
```

执行时得：

```
yt2 =
1 - 1/2 * i/r * (1/( - xi + i * r) * exp(( - xi + i * r) * t) - 1/( - xi - i * r) * exp(( - xi - i * r) * t))
```

为使该表达式进一步简化，使用 subs 做以下代换：

```
exp(( - xi + i * r) * t)→exp( - xi * t) * (cos(r * t) + i * sin(r * t))
exp(( - xi - i * r) * t)→exp( - xi * t) * (cos(r * t) - i * sin(r * t))
```

再对结果使用 simple 化简。程序变为：

```
syms s t
syms xi r real
den = s * (s^2 + 2 * xi * s + 1);
yt = ilaplace(1/den,s,t);
yt1 = subs(yt,'(4 * xi^2 - 4)^(1/2)','2 * j * r');
yt2 = simple(yt1);
yt3 = subs(yt2,'exp(( - xi + i * r) * t)','exp(( - xi * t) * (cos(r * t) + i * sin(r * t))');
yt4 = subs(yt3,'exp(( - xi - i * r) * t)','exp(( - xi * t) * (cos(r * t) - i * sin(r * t))');
yt5 = simple(yt4)
```

执行程序得：

```
yt5 =
(r * xi^2 + r^3 - exp( - t * xi) * cos(r * t) * r - exp( - t * xi) * sin(r * t) * xi)/(r * xi^2 + r^3)
```

由于：

$$\text{xi\^{}2} + \text{r\^{}2} \rightarrow \xi^2 + (\sqrt{1-\xi^2})^2 = 1$$

因此，有：

$$\text{r * xi\^{}2 + r\^{}3} \rightarrow r$$

然后，用 simple 对结果进行化简。最后的程序为：

```
syms s t
syms xi r real
den = s * (s^2 + 2 * xi * s + 1);
yt = ilaplace(1/den,s,t);
yt1 = subs(yt,'(4 * xi^2 - 4)^(1/2)','2 * j * r');
yt2 = simple(yt1);
yt3 = subs(yt2,'exp(( - xi + i * r) * t)','exp(( - xi * t) * (cos(r * t) + i * sin(r * t))');
yt4 = subs(yt3,'exp(( - xi - i * r) * t)','exp(( - xi * t) * (cos(r * t) - i * sin(r * t))');
yt5 = simply(yt4);
yt6 = subs(yt5,'r * xi^2 + r^3','r');
yoft = simple(yt6)
```

执行程序得：

```
yoft =
1 - 1/exp(t * xi) * cos(r * t) - 1/r/exp(t * xi) * sin(r * t) * xi
```

将结果手工转换为标准代数形式，有：

$$y(t) = 1 - e^{-\xi t}\left[\cos(t\sqrt{1-\xi^2}) + \frac{\xi}{\sqrt{1-\xi^2}}\sin(t\sqrt{1-\xi^2})\right] \tag{5.4}$$

例 5.18 卷积计算及其特征

由式(5.3)的解得到的卷积如下式,其中,$\xi<1$,$h(t)=u(t)$,$u(t)$为单位阶跃函数:

$$y(t)=\frac{e^{-\xi_t}}{\sqrt{1-\xi^2}}\int_0^t e^{\xi\eta}\sin[(t-\eta)\sqrt{1-\xi^2}]d\eta$$

首先,求出该积分的符号解,将其转换为函数,并绘出图形;其次,在 $dy/dt=0$ 时求出其最大值及对应的时间,并用二阶导数验证是否为最大值;第三,求出 $y(t)$ 由 $0.1y(t)$ 变到 $0.9y(t)$ 时所需的时间,即信号的上升时间。

第一部分的程序为:

```
option = optimset('Display','off');
z = 0.15;
syms t xi n r a
r = sqrt(1 - xi^2);
arg = exp(xi * n) * sin(r * (t - n));
yt = exp( - xi * t) * int(arg,n,0,t)/r;
yoft = inline(vectorize(yt),'t','xi');
tt = linspace(0,20,200);
plot(tt,yoft(tt,z))
```

执行后产生图 5.14。检查符号表达式 yt 将发现它与式(5.4)相等。

为求最大值,对解 $y(t)$求微分,然后转换为函数,使用 fzero 求导数为 0 时的时间 t_{max},再通过求 $y(t)$的二阶导数来证实最大值。如果在 t_{max}处 $y(t)$的二阶导数为负,则确定函数为最大值。修改后的程序为:

```
option = optimset('display','off');
z = 0.15;
syms t xi n r a
r = sqrt(1 - xi^2);
arg = exp(xi * n) * sin(r * (t - n));
yt = exp( - xi * t) * int(arg,n,0,t)/r;
yoft = inline(vectoroize(yt),'t','xi');
tt = linspace(0,20,200);
plot(tt,yoft(tt,z))
% Part (2)
dydt = inline(vectorize(diff(yt,t)),'t','xi');
tmax = fzero(dydt,[3 5],option,z);
ymax = yoft(tmax,z);
disp(['ymax = 'num2str(ymax)'tmax = 'num2str(tmax)])
d2ydt2 = inline(vextorize(diff(yt,t,2)),'t','xi');
secder = d2ydt2(tmax,z);
disp(['Second derivative at tmax = 'num2str(secder)])
```

执行程序后在命令窗口显示:

```
ymax = 1.6209   tmax = 3.1775
Second derivative at tmax = -0.62087
```

通过创建一个新的 inline 函数计算 $y(t)-a$ 来求得上升时间,$a=0.1$ 或 0.9。程序为:

```
option = optimset('Display','off');
z=0.15;
syms t xi n r a
r= sqrt(1 - xi^2);
arg = exp(xi * n) * sin(r * (t - n));
yt = exp( - xi * t) * int(arg,n,0,t)/r;
yoft = inline(vectoroize(yt),'t','xi');
tt = linspace(0,20,200);
plot(tt,yoft(tt,z))

% Part (2)

dydt = inline(vectorize(diff(yt,t)),'t','xi');
tmax = fzero(dydt,[3 5],option,z);
ymax = yoft(tmax,z);
disp(['ymax = 'num2str(ymax)'tmax = 'num2str(tmax)])
d2ydt2 = inline(vextorize(diff(yt,t,2)),'t','xi');
secder = d2ydt2(tmax,z);
disp(['Second derivative at tmax = 'num2str(secder)])

% Part (3)

ytrise = inline(vectorize(yt - a),'t','xi','a');
t9 = fzero(ytrise,[0 2],option,z,0.9);
t1 = fzero(ytrise,[0 2],option,z,0.1);
disp(['Rise time = 'num2str(t9 - t1)])
```

执行程序后在命令窗口显示：

```
ymax = 1.6209    tmax = 3.1775
Second derivative at tmax = -0.62087
Rise time = 1.1518
```

5.7 应用函数一览

表 5.3 对本章介绍的函数及其说明做了总结。表 5.4 对符号工具箱的函数做了总结。

表 5.3 第 5 章介绍的 MATLAB 函数

MATLAB 函数	说明
bvp4c	对具有两点边界值的常微分方程求解
bvpinit	形成 bvp4c 函数的初始估计值
conv	两多项式相乘
deval	求 bvp4c 的解
dblquad	对双积分进行数值计算
diff	求数组内相邻两元素的差
error	显示错误信息
feval	计算函数值
fft	求离散傅里叶变换
fminbnd	函数的自变量在特定间隔内求其极小值
fsolve	解非线性方程(优化工具箱)

(续表)

MATLAB 函数	说明
function	创建函数的m文件
fzero	找到有一个变量的函数的零点
global	定义全局变量
humps	MATLAB演示函数
ifft	求离散反傅里叶变换
inline	构建 inline 函数
interp1	执行一维插值运算
nargin	求函数接口中的参数个数
ode45	对初始值问题的常微分方程求解(6个中的一个)
odeset	在常微分方程求解函数中改变选择
optimset	在包括 fzero 和 fminbnd 的优化函数中改变选择
poly	由多项式的根创建多项式
polyfit	n阶多项式的拟合
polyval	计算多项式的值
quadl	对一阶积分的数值进行计算
return	实现函数提前返回
roots	求多项式的根
spline	用 spline 函数实现数据拟和
trapz	用梯形估计法计算一阶积分

表 5.4 第 5 章介绍的符号工具箱中的 MATLAB 函数

MATLAB 函数	说明
diff	针对符号变量求微分
ilaplace	针对符号变量求反拉普拉斯变换
int	针对符号变量求积分
limit	求符号表达式的极限
simple	求符号表达式的最简式
simplify	对符号表达式进行化简
subs	对符号表达式进行代换
syms	创建符号变量的捷径
vectorize	将表达式转换为字符串表达式
vpa	对符号表达式使用变量精确计算

练习

5.1 如图 5.15 所示,阶跃循环轴的应力集中系数可由下式近似估算[①]:

$$K_t = c\left(\frac{D-d}{2d}\right)^{-a}$$

其中,c 和 a 在表 5.5 中给出。通过以下两种方法求取 c 和 a 的两个表达式(D/d 的函数):(1)用五阶多项式;(2)用 spline 函数。对于上述两种方法,比较由两组拟合值获得的 K_t 值和表 5.5 中给出的初始值。指出哪一个是这种情况下较好的一种方式。

① 参见 R. L. Norton, *Machine Design: An Integrated Approach*, Prentice Hall, Upper Saddle River, NJ, 1996 年,第 1005ff 页。

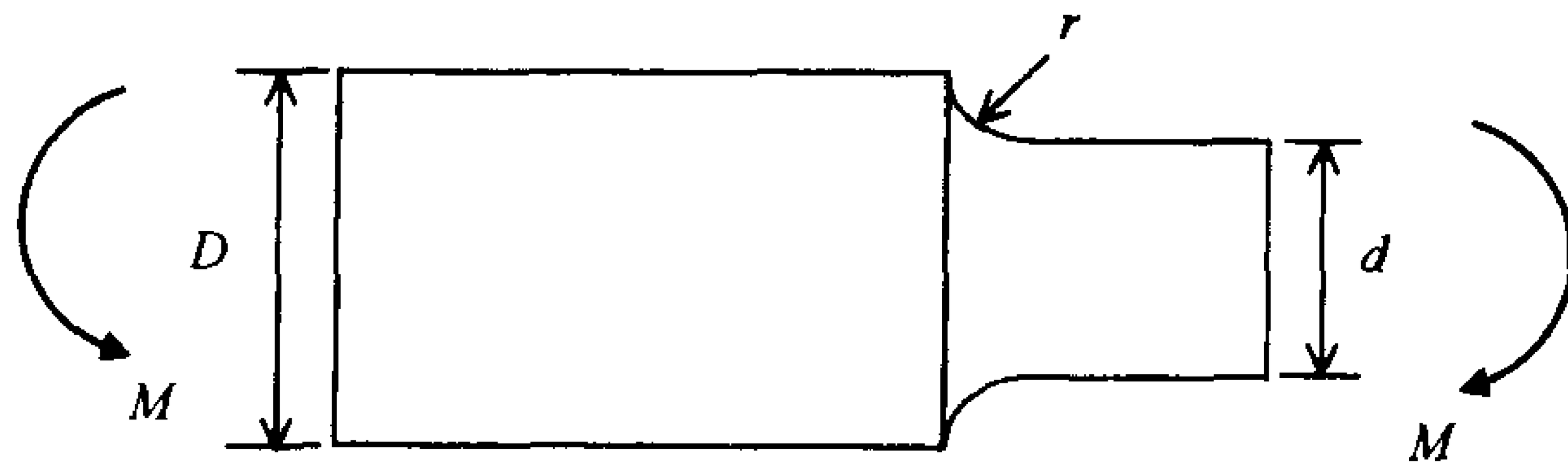


图5.15　应力集中系数的几何图形和负荷

表5.5　应力集中系数常数

D/d	c	a
6.00	0.88	0.33
3.00	0.89	0.31
2.00	0.91	0.29
1.50	0.94	0.26
1.20	0.97	0.22
1.10	0.95	0.24
1.07	0.98	0.21
1.05	0.98	0.20
1.03	0.98	0.18
1.01	0.92	0.17

5.2　考虑以下信号：

$$f(t)=\sum_{n=1}^{4}H_n e^{-\zeta_n\omega_n t}\sin\left(\sqrt{1-\zeta_n^2}\omega_n t\right)\qquad 0\leqslant t\leqslant T$$

其中，常数值由表5.6给出。对于 $N=2^{10}$ 和 $\Delta t=2\pi/(4\omega_4)$：

(a) 分别对带有和不带有汉明权函数的信号画出幅度频谱。结果如图5.16所示，但无标注。参考第6章在图中加坐标轴标注和文字的方法。

(b) 确定峰值出现时的频率。[提示：用 find 和 diff 的几个应用。]

[答案：不带有汉明权函数：[4.84375,9.14063,20.0781]Hz，带有汉明权函数：[4.92188,9.0625,9.45313,20.0781]Hz。]

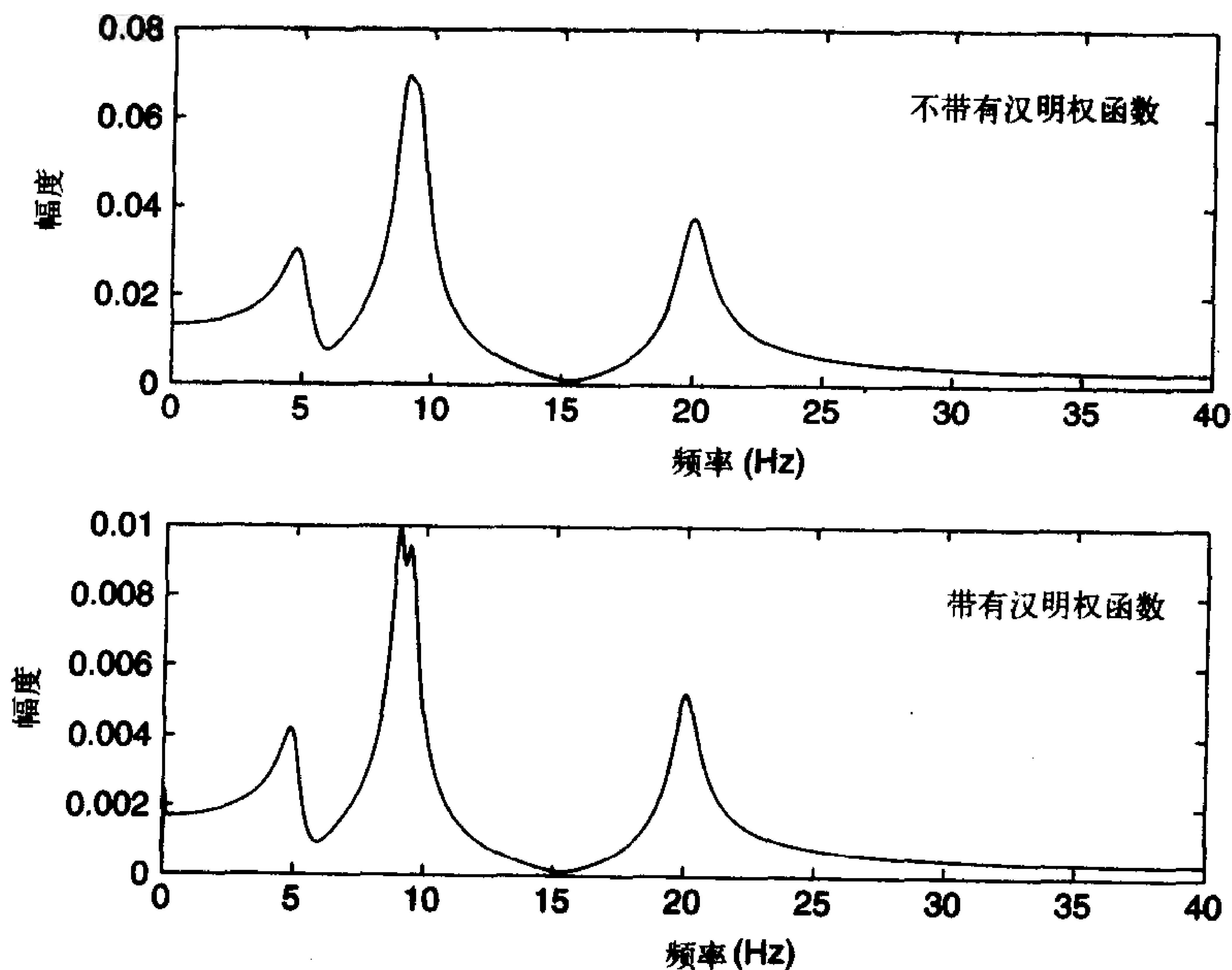


图5.16　习题5.2(a)的解

表5.6 定义习题5.2中信号的常数

n	$\omega_n/2\pi$	ζ_n	H_n
1	5	0.1	1
2	9	0.04	1.3
3	9.4	0.04	1.3
4	20	0.03	1.8

5.3 由下面多项式的根确定主应力[1]:

$$\sigma^3 - C_2\sigma^2 - C_1\sigma - C_0 = 0$$

其中,

$$C_2 = \sigma_x + \sigma_y + \sigma_z$$
$$C_1 = \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2 - \sigma_x\sigma_y - \sigma_y\sigma_z - \sigma_z\sigma_x$$
$$C_0 = \sigma_x\sigma_y\sigma_z + 2\tau_{xy}\tau_{yz}\tau_{zx} - \sigma_x\tau_{yz}^2 - \sigma_y\tau_{zx}^2 - \sigma_z\tau_{xy}^2$$

$\sigma_x,\sigma_y,\sigma_z$ 为施加的法向应力,$\tau_{xy},\tau_{yz},\tau_{zx}$ 为施加的切向应力。如果方程的根是三个主应力 σ_1,σ_2 和 σ_3,且 $\sigma_1 > \sigma_2 > \sigma_3$,则切向主应力为:

$$\tau_{12} = (\sigma_1 - \sigma_2)/2 \qquad \tau_{23} = (\sigma_2 - \sigma_3)/2 \qquad \tau_{13} = (\sigma_1 - \sigma_3)/2$$

其中,$\tau_{max} = \tau_{13}$。

确定主应力值,即可得到相应的切向应力值:

$$\sigma_x = 100 \qquad \tau_{xy} = -40$$
$$\sigma_y = -60 \qquad \tau_{yz} = 50$$
$$\sigma_z = 80 \qquad \tau_{zx} = 70$$

寻根函数未对根进行排序。可以使用 sort 函数按照2.5.4节例2.2所讨论的方式对根进行排序。

[答案:$\sigma_1 = 160.7444, \sigma_2 = 54.8980, \sigma_3 = -95.6424, \tau_{12} = 52.9232, \tau_{23} = 75.2702, \tau_{13} = 128.1934$。]

注意:在习题5.4至习题5.16中,要求用 fzero 函数求取下列方程最接近的五个正根,特殊声明除外。函数仅要求在$[x_0\ x_1]$区域内进行搜索。确定根以前绘制函数图形,可使用 axis 提高图形分辨率(参见6.2.2节)。

5.4 弹簧的振动方程[2]:

$$\tan x = x$$

5.5 平板的对流换热方程[3]。$p = 0.1$ 和 $p = 1$ 时分别求取方程的根。

$$2\cot x = \frac{x}{p} - \frac{p}{x}$$

5.6 环形薄膜的振动方程[4],设 $b = 2$。

$$J_0(x)Y_0(xb) - J_0(xb)Y_0(x) = 0$$

用 besselj 和 bessely 分别求取一类零阶贝塞尔函数 $J_0(x)$和二类零阶贝塞尔函数 $Y_0(x)$。

---

① 参见 J.E.Shigley 和 C.R.Mischke, *Mechanical Engineering Design*,第五版,McGraw-Hill,New York,1998年。
② 参见 E.B.Magrab, *Vibration of Elastic Structural Members*, Sijthoff & Noordhoff, The Netherlands, 1979年,第58页。
③ 参见 M.N.Ozisik, *Heat Conduction*,第二版,John Wiley & Sons,New York, 1993年,第47页。
④ 参见 E.B.Magrab,出处同上,第83页。

5.7 一根悬臂梁，自由端附加集中质量 M_0，其振动方程[⑤]如下所示。当 $M_0/m_0=0,0.2$ 和 1 时分别求根。

$$(M_0/m_0)\Omega[\cos(\Omega)\sinh(\Omega)-\sin(\Omega)\cosh(\Omega)]+\cos(\Omega)\cosh(\Omega)+1=0$$

5.8 一端夹紧，另一端支撑的简支梁振动方程[⑥]为：

$$\tanh(\Omega)-\tan(\Omega)=0$$

5.9 边沿夹紧的固体圆盘的振动方程[⑦]如下：

$$J_m(\Omega)I_{m+1}(\Omega)+I_m(\Omega)J_{m+1}(\Omega)=0$$

其中，$J_m(x)$是一类 m 阶贝塞尔函数，$I_m(x)$是一类 m 阶贝塞尔修正函数。分别用 besselj 和 besseli求 $J_m(x)$和 $I_m(x)$。求取 $m=0,1$ 和 2 时最接近的三个根。

5.10 下面的方程[⑧]来源于悬索对称模式的求取。求解当 $\lambda^2=2\pi^2,4\pi^2$ 和 $8\pi^2$ 时的最接近根。这一解必须通过交替画图得到。用 axis 函数将纵轴限制在从 -10 到 20 的范围内。

$$\tan\Omega=\Omega-\frac{4\Omega^3}{\lambda^2}$$

5.11 分析梯形截面开口管道内不均匀流动时，流体深度与能量梯度 x 的高度比由下式确定[⑨]。

$$(1+c_0x)^2(x^2-x^3)=c_1$$

其中，$0\le c_0\le 11$ 和 $0.005\le c_1\le 12.3$ 是管道的几何形状和流速的函数。然而，并不是所有 c_0 和 c_1 的组合都是合适的。在 0 和 1 之间求一对实数值 x，并在：(1) $c_0=0.4, c_1=0.2$；(2) $c_0=7.0, c_1=4.0$ 的条件下使方程成立。用fzero和roots两种方法求解方程。用 roots 求解时方程重写为：

$$-c_0^2x^5+(c_0^2-2c_0)x^4+(2c_0-1)x^3+x^2-c_1=0$$

5.12 在开口管道流体顶部的扰动波内，流体的流速大于波动速度，波动角$\beta(0<\beta\le\pi/2)$由下式确定[⑩]。

$$2N_F^2\sin^2(\beta)\tan^2(\beta-\theta)=\tan(\beta)\tan(\beta-\theta)+\tan^2(\beta)\qquad \beta>\theta$$

其中，θ 为墙偏转角，$1\le N_F\le 12$ 为弗劳德(Froude)数。当 $\theta=35°, N_F=5$ 时，求 β 值$(0<\beta\le 90°)$。首先，画出函数图形。

5.13 内部收益率 i_{rr} 作为投资结余所获利润的百分比，可由下式[⑪]确定。

$$\sum_{k=0}^{n}F_k(1+i_{rr})^{-k}=0$$

其中，n 是期数，i_{rr}是以十进制表示的内部收益率，F_k是每期的现金流量：正的现金

---

⑤ 参见 E.B.Magrab，出处同上，第 130 页

⑥ 参见 E.B.Magrab，出处同上，第 130 页。

⑦ 参见 E.B.Magrab，出处同上，第 252 页。

⑧ 参见 M.Irvine, *Cable Structures*, Dover Publications, New York, 1981 年，第 95 页。

⑨ 参见 H.W.King, *Handbook of Hydraulics*, 第四版, McGraw-Hill, New York, 1954 年，第 81 页。

⑩ 参见 N.H.C.Hwang 和 C.E.Hita, *Fundamentals of Hydraulic Engineering Systems*, 第二版, Prentice Hall, Englewood Cliffs, NJ, 1987 年，第 222 页。

⑪ 参见 G.J.Theusen 和 W.J.Fabrycky, *Engineering Economy*, 第八版, Prentice Hall, Englewood Cliffs, NJ, 1993 年，第 176 页。

流量表示现金吸纳;负的现金流量表示现金支付。当 $F_0 = -\$1000$, $F_1 = -\$800$, $F_2 = \$500$, $F_3 = \$500$, $F_4 = \$500$, $F_5 = \$1200$ 时,求 i_{rr}。

5.14 如果投资总额为 P,该期内收益额为 A,则每期利率为 i 时,P 的投资回收期 n 由下式确定①。

$$\frac{A}{P} = \frac{i(1+i)^n}{(1+i)^n - 1}$$

如果 $i = 12\%$ /年,$A/P = 0.16$,则确定 n,即收回投资的年数。

5.15 韦布尔(Weibull)概率密度函数中参数 β 的估计值(参见 14.2.2 节)由以下方程的解确定②:

$$\beta = \left[\frac{\sum_{i=1}^{n} x_i^{\beta}\ln(x_i)}{\sum_{i=1}^{n} x_i^{\beta}} - \frac{1}{n}\sum_{i=1}^{n}\ln(x_i)\right]^{-1}$$

其中,x_i是大小为 n 的随机采样。如果

x = [72 82 97 103 113 117 126 127 127 139 154 159 199 207]

确定 β 值。

5.16 在确定一个球体和平面的表面接触剪切应力时(是轴承相对于表面的影响模型),比例 x 的值由下式获得③。

$$x\ln(\sqrt{x^2-1} + x) - \sqrt{x^2-1} - Cx = 0$$

其中,$x > 1$, $C < 1$。当 $C = 0.5$ 时,确定 x 值。

5.17 求解下列方程的三个实根④:

$$x^4 = 2^x$$

[提示:首先在 $-1 \leqslant x \leqslant 2$ 和 $2 \leqslant x \leqslant 17$ 两个不同区域中绘制函数图形。]

5.18 某气体的可压缩性系数为 Z,其广义方程的计算公式由下式给出⑤:

$$\begin{aligned} Z(r,\tau) = {} & 1 + r\sum_{i=1}^{6} A_i\tau^{i-1} + r^2\sum_{i=7}^{10} A_i\tau^{i-7} + r^3\sum_{i=11}^{13} A_i\tau^{i-11} + r^4 A_{14}\tau \\ & + r^5(A_{15}\tau^2 + A_{16}\tau^3) + r^6 A_{17}\tau^2 + r^7(A_{18}\tau + A_{19}\tau^3) \\ & + r^8 A_{20}\tau^3 + r^2 e^{-0.0588r^2}[A_{21}\tau^3 + A_{22}\tau^4 + r^2(A_{23}\tau^3 + A_{24}\tau^5) \\ & + r^4(A_{25}\tau^3 + A_{26}\tau^4) + r^6(A_{27}\tau^3 + A_{28}\tau^5) + r^8(A_{29}\tau^3 + A_{30}\tau^4) \\ & + r^{10}(A_{31}\tau^3 + A_{32}\tau^4 + A_{33}\tau^5)] \end{aligned}$$

其中,$\tau = T_c/T(0.4 \leqslant \tau \leqslant 1)$; $r = RT_c/P_c v$; R 为气体常数,单位为(MPa-m³)/(kg-K); T 为温度,单位为 K; P 为压强,单位为 MPa; v 为体积,单位为 m³/kg; T_c和P_c分别为

① 参见 G.J.Theusen 和 W.J.Fabrycky,出处同上,第 188 页。

② 参见 D.C.Montgomery 和 G.C.Runger, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, New York, 1994 年,第 299 页。

③ 参见 W.Changsen, *Analysis of Rolling Element Bearings*, Mechanical Engineering Publishers, London, 1991,第 80 页。

④ 参见 Problem suggested by Prof. Jeffery M. Cooper, Department of Mathematics, University of Maryland, College Park, MD。

⑤ 参见 W.C.Reynolds, "Thermodynamic Properties in SI," Department of Mechanical Engineering, Stanford University, Stanford, CA, 1979 年。

临界温度和临界压力,33 个常数值由表 5.7 给出。

(a) 创建一个函数以确定 $Z(r,\tau)$。用下列检验值检查函数,并使用 `format long e` 选项:

(i) Z(1,1) = 0.702 423 969 27

(ii) Z(1/0.3,1) = 0.299 999 999 80

(iii) Z(2.5,0.5) = 0.992 218 539 28

(b) 上面各量应用于下式:

$$Z(r,\tau) = \frac{p\tau}{r} = \frac{Pv}{RT} \tag{a}$$

其中,$p = P/P_c$ $(1 \leqslant p \leqslant 6)$。使用式(a)在下列条件下确定 r 和 $Z(r,\tau)$的值:(i)$p = 0.6$, $\tau = 1/1.05$;(ii)$p = 2.18$, $\tau = 1/1.2$。

[答案:(i) $r = 0.7131$ 时 $Z = 0.8013$;(ii) $r = 3.3567$ 时 $Z = 0.5412$。]

(c) 使用式(a)在下列条件下确定 τ 和 $Z(r,\tau)$的值:(i)$p = 0.6$, $r = 1/1.4$;(ii)$p = 2.18$, $r = 1/0.6$。

[答案:(i)$\tau = 0.9532$ 时 $Z = 0.8007$;(ii)$\tau = 0.6505$ 时 $Z = 0.8508$。]

表 5.7　广义公式中 Z 的常数

j	A_j	j	A_j	j	A_j
1	0.062 432 384	12	−0.000 727 155 024 313	23	−0.084 519 449 381 3
2	0.127 214 77	13	−0.004 524 546 526 10	24	−0.003 409 313 119 28
3	−0.936 332 33	14	0.001 304 687 241 00	25	−0.001 951 270 499 01
4	0.701 844 11	15	−0.000 222 165 128 409	26	$4.938\,999\,109\,78 \times 10^{-5}$
5	−0.351 608 96	16	−0.001 981 405 356 56	27	$-4.932\,646\,129\,30 \times 10^{-5}$
6	0.056 450 032	17	$5.975\,739\,729\,21 \times 10^{-5}$	28	$8.856\,665\,723\,82 \times 10^{-7}$
7	0.029 956 146 990 7	18	$-3.641\,353\,497\,02 \times 10^{-6}$	29	$5.347\,880\,295\,53 \times 10^{-8}$
8	−0.031 817 436 764 7	19	$8.413\,648\,453\,86 \times 10^{-6}$	30	$-5.934\,205\,591\,92 \times 10^{-8}$
9	−0.016 821 105 551 7	20	$-9.828\,688\,588\,22 \times 10^{-9}$	31	$-9.068\,133\,269\,29 \times 10^{-9}$
10	1.602 040 600 81	21	−1.576 830 568 10	32	$1.618\,224\,072\,65 \times 10^{-9}$
11	−0.001 099 967 407 46	22	0.040 072 898 890 8	33	$-3.320\,447\,939\,15 \times 10^{-10}$

5.19 作为管道摩擦系数 λ 的函数,管道中流动的流体压降可由柯尔勃洛克(Colebrook)公式进行估算①:

$$\lambda = \left[-2\log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{0.27}{d/k}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

其中,R_e是雷诺数,d 是管道直径,k 是表面粗糙度。对于光滑管道($k \cong 0$ 或 $d/k >$ 100 000):

$$\lambda = \left[-2\log_{10}\left(\frac{R_e\sqrt{\lambda}}{2.51}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

对于充分掺混的紊流流动而言,摩擦系数由下式给定:

$$\lambda = \left[2\log_{10}\left(3.7\,\frac{d}{k}\right)\right]^{-2}$$

① 参见 N.H.C.Hwang 和 C.E.Hita,出处同上,第 68 页。

独立于 R_e。该式是柯尔勃洛克通用公式的一个特例,用于获得迭代类型问题的初值,如 11.2.3 节所述。

如果 λ 值的范围从 0.008 到 0.08,则当 $R_e = 10^5$,(1) $d/k = 200$,(2) $k = 0$ 时求 λ 值。

[答案:(1)$\lambda = 0.0313$;(2)$\lambda = 0.0180$。]

5.20 求下列多项式的实根并显示。

$$10x^6 - 75x^3 - 190x + 21 = 0$$

5.21 用quadl 和 trapz求取图 5.17 所示的两条正弦曲线之间的面积。两个正弦函数分别为 $\sin(x)$和$|\sin(2x)|/2$。

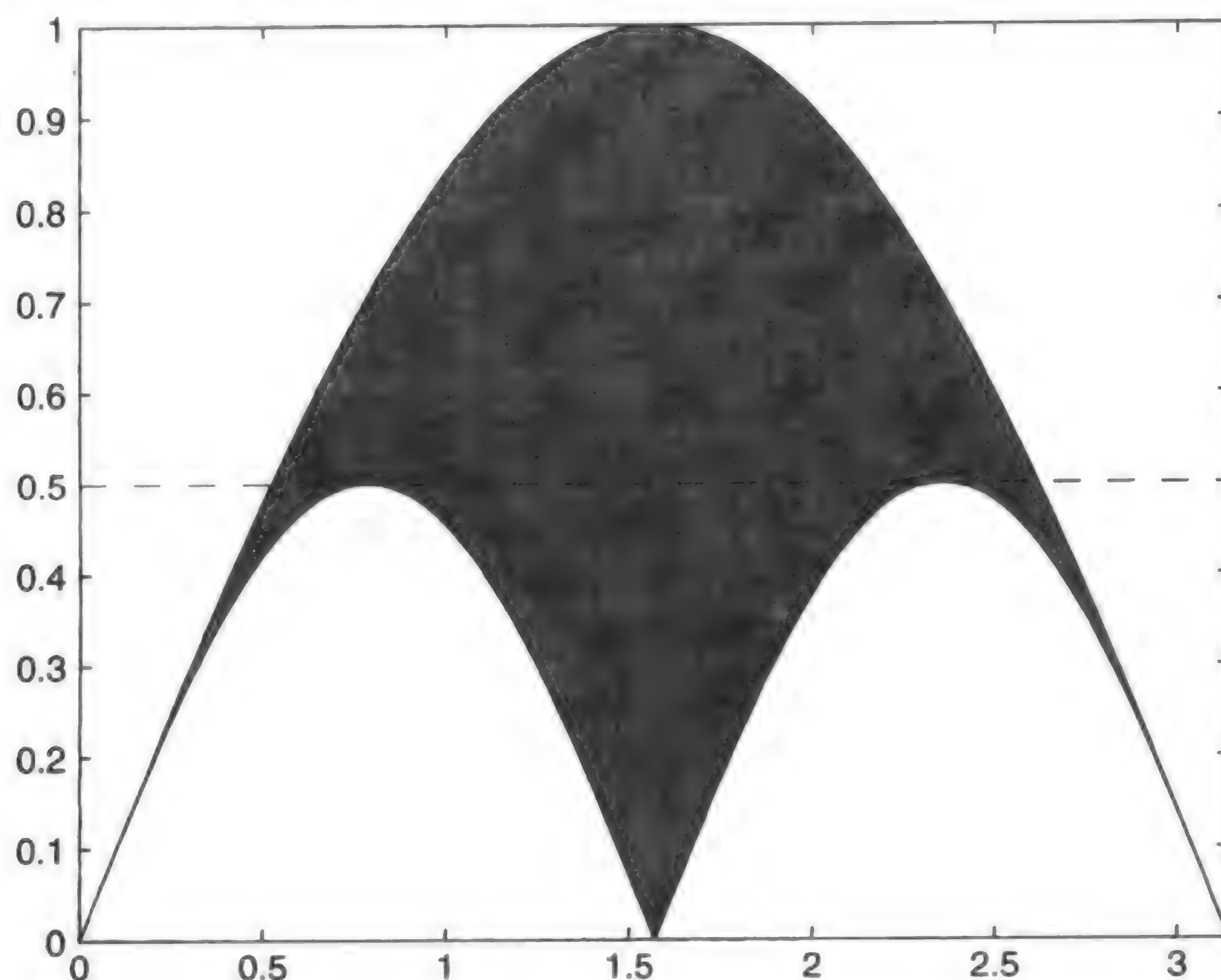


图 5.17 习题 5.21 的示图

5.22 为确定偏心负荷下轴向推力轴承的负荷分布,必须进行以下的积分运算①:

$$I_m(\varepsilon) = \frac{1}{2\pi}\int_{-a}^{a}[1-(1-\cos(x)]2\varepsilon^c\cos(mx)\mathrm{d}x$$

其中,$\varepsilon > 0$,$m = 0$ 或 1,且:

$$a = \arccos(1-2\varepsilon)$$

滚珠轴承 $c = 1.5$,滚柱轴承 $c = 1.1$。确定滚珠轴承 $I_1(0.6)$的值。

[答案:$I_1(0.6) = 0.2416$。]

5.23 给定:

$$\int_0^\infty E_{\lambda,b}(\lambda,T)\mathrm{d}\lambda = \sigma T^4$$

其中,

$$E_{\lambda,b}(\lambda,T) = \frac{C_1}{\lambda^5[\exp(C_2/\lambda T)-1]}$$

λ 为波长,单位为 μm;T 为温度,单位为 K;$C_1 = 3.742\times10^8$ W·μm^4/m^2;$C_2 = 1.439$

① 参见 W. Changsen,出处同上,第 92 页。

$\times 10^4$ μm·K；$\sigma = 5.667 \times 10^{-8}$ W/m^2·K^{4}为斯蒂芬－波耳兹曼（Stephan-Boltzmann）常数。$T = 300$ K，400 K和500 K时进行数值积分，并计算结果的相对误差。两个积分限很难用数字量给出，因而近似积分时采用下限1 μm和上限150 μm。积分限由上述三个温度值处的$E_{\lambda,b}$图和不引起quadl产生警告信息的积分限值确定。

[答案：$error_{300} = 0.145\%$，$error_{400} = 0.061\%$，$error_{500} = 0.030\%$。]（可参见习题12.6。）

5.24 计算以下积分：

$$Z = \int_{\pi/4}^{\pi}\int_{0}^{\pi/2} \cos(x-y)e^{-xy/\pi^2}\,dx\,dy$$

5.25 一枚导弹，以初始速度v_0，水平夹角α离开原点(0,0)。如果导弹在(x_e, y_e)点着陆，且在飞行中受到一拉力，其大小和速度的平方成比例，则控制导弹飞行轨迹的四个一阶方程为①：

$$\frac{dv_x}{dx} = -c_d v \qquad \frac{dv_y}{dx} = \frac{-(g + c_d v v_y)}{v_x} \qquad \frac{dy}{dx} = \frac{v_y}{v_x} \qquad \frac{dt}{dx} = \frac{1}{v_x}$$

其中，y是导弹的垂直高度，x是飞行的水平距离，t是时间，v_x和v_y分别是速度v的水平和垂直分量。c_d是拉力系数，g是重力加速度，且：

$$v = \sqrt{v_x^2 + v_y^2}$$

仅当v_0足够大，以至于当到达x_e，v_x大于零时这些方程才有效。条件测试可以判断$|v_x| > v_0 \times 10^{-6}$的状态。如果条件不满足，则必须终止程序的执行。终止操作用error实现。条件检查应放置在ode45函数调用的开始处。初始条件为：

$$v_{0x} = v_0\cos(\alpha) \qquad v_{0y} = v_0\sin(\alpha) \qquad y = 0 \qquad t = 0$$

令$y_1(x) = v_x$，$y_2(x) = v_y$，$y_3(x) = y$，$y_4(x) = t$，依次写出各方程。

(a) 当$v_0 = 600$ fps，$c_d = 0.002$，$\alpha = 45°$时，画出导弹到达$y_e = 0$前的飞行轨迹，即在$y_e > 0$的条件下绘图。ode45中令$x_{final} = 1000$ ft。

(b) 求导弹的最大上升高度及出现这一高度时的飞行距离。用fminbnd和spline函数求解。

[答案：$x = 648.1205$ ft时$y_{max} = 474.8285$ ft。]

(c) 求$y_e = 0$时的x_e值及到达这一点时所需的时间。用interp1求解。

[答案：$x_e = 975.3240$，所需时间为10.6246 s。]

5.26 一位蹦极爱好者准备从一高空热气球跳下，所用橡皮带长度为L。为保证安全，必须预知最大加速度、速度和总下落高度，确保使力不会太大，而且气球足够高以保证蹦极者不会撞到地面。考虑空气动力学阻力，控制方程为②：

$$\frac{d^2x}{dt^2} + c_o\,\text{signum}(dx/dt)\left(\frac{dx}{dt}\right)^2 + \frac{k}{m_j}(x-L)u(x-L) = g$$

其中，$g = 9.8$ m/s^{2}为重力加速度；c_d和阻力系数成比例，单位为m^{-1}；k为橡皮带的

① 参见H.B.Wilson和L.H.Turcotte，*Advanced Mathematics and Mechanics Applications Using MATLAB*，第二版，CRC Press，Boca Raton，FL，1997年，第294页。

② 参见D.M.Etter，*Engineering Problem Solving with MATLAB*，Prentice Hall，Upper Saddle River，NJ，1997年，第220～221页。

弹性系数,单位为 N/m; m_j 为蹦极者的质量; $u(z)$ 为单位阶跃函数,即 $z\leqslant 0$ 时, $u(z)=0$; $z>0$ 时, $u(z)=1$。如果采用 4.1 节所述的逻辑运算符描述 $u(z)$,则程序将大为简化。

如果 $L=150$ m, $m_j=70$ kg, $k=10$ N/m, $c_o=0.00324\ \text{m}^{-1}$,初始条件为零,试证明:

(1)11.47 s 时, x 的最大值为 -308.47 m。

(2)5.988 s 时,蹦极者将下降 150 m,速度为 -43.48 m/s。

(3)11.18 s 时,最大加速度将是 $-12.82\ \text{m/s}^2(-1.308\ g)$。

画出位移、速度和加速度曲线。加速度由diff求得的速度的近似导数获得。对ode45的输出应用spline可得到上述数值解。

5.27 一个由长度为 L 的无重量刚性棒组成的倒置摆,其自由末端系有一个质量为 m 的物体和一个弹性常数为 k 的线性弹簧。开始时钟摆处于垂直位置,且未拉伸时,弹簧的长度为 L。摆轴的转动阻尼为 c,摆由力矩 $M(t)$ 驱动。则描述倒置摆角运动的控制方程为①:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\tau^2}+\alpha\frac{\mathrm{d}\theta}{\mathrm{d}\tau}-\sin\theta+\beta\left(1-\frac{1}{\sqrt{5-4\cos\theta}}\right)\sin\theta=P(t)$$

其中,

$$\beta=\frac{2kL}{mg}\qquad P=\frac{M}{mgL}\qquad \tau=t\sqrt{\frac{g}{L}}\qquad \alpha=(c/m)\sqrt{L/g}$$

t 为时间。

如果 $M=0,\beta=10,\alpha=0.1,\theta(0)=\pi/4,\mathrm{d}\theta(0)/\mathrm{d}\tau=0$,画出转角 θ 的曲线(θ 为 τ 的函数,且 τ 在 $0\leqslant\tau\leqslant 50$ 范围内 1000 等分),并在一单独图表中画出 $\theta(\tau)$ 相对于 $\mathrm{d}\theta(\tau)/\mathrm{d}\tau$ 的图形。

5.28 由一长管连接的两个矩形等截面水库的高度差为 Z,所产生的振动由下式确定②:

$$\frac{\mathrm{d}^2Z}{\mathrm{d}t^2}+\mathrm{signum}(\mathrm{d}Z/\mathrm{d}t)p\left(\frac{\mathrm{d}Z}{\mathrm{d}t}\right)^2+qZ=0$$

如果 $p=0.375\ \text{m}^{-1}, q=7.4\times10^{-4}\ \text{s}^{-2}$,初始条件为 $Z(0)=Z_n$ m, $\mathrm{d}Z(0)/\mathrm{d}t=0$ m/s,求当 $Z_1=10$ m, $Z_2=50$ m 时,第一次出现 $Z(t_n)=0$ 时的 $t_n(n=1,2)$ 值。用interp1求 t_n,数值符号由sign确定。建议:绘制一幅 Z_n 值的结果图,然后根据曲线的特性,使用min 和 find的适当组合来选择小范围值的中值索引, interp1可借助其实现插值。

[答案: $t_1=114.2692$ s, $t_2=276.1428$ s。]

5.29 考虑式(5.3)及其在输入为阶跃函数时的数值解,即输入 $h(t)=u(t)$。求使下式中的 $f(\xi)$ 最小时的 ξ 值:

$$f(\xi)=\sum_{n=1}^{N}(y(t_n)-1)^2$$

ξ 由 0.05 变到 0.95, 每次增加 0.05。这里应注意, $f(\xi)$ 为数值数组,因而不能使

① 参见 H. B. Wilson 和 L. H. Turcotte,出处同上,第 279 页。

② 参见 D. N. Roy, *Applied Fluid Mechanics*, Ellis Horwood Limited, Chichester, England, 1988 年,第 290~293 页。

用fminbnd求解。使用min找回ξ的索引及该索引对应的ξ值。

5.30 一根均匀不可伸展的电缆，长度为L_o，单位长度的重量为w，悬挂在两个固定点$x=0$和$x=L$之间($L<L_o$)。如果绳子没有任何弯曲刚度，且仅能支持拉力T，则电缆的无量纲挠度$z(\eta)$的控制方程为[①]：

$$\frac{d^2z}{d\eta^2}=\beta\sqrt{1+\left(\frac{dz}{d\eta}\right)^2}$$

其中，$\eta=x/L$，$\beta=wL/H$，H是T的水平分量，负z表明向下偏移。相应电缆的长度L_o等于：

$$L_o=L\int_0^1\sqrt{1+\left(\frac{dz}{d\eta}\right)^2}d\eta$$

当给出w，L和L_o时可确定β和H值。其边界条件为：

$$z(0)=0\qquad z(1)=0$$

$L_o/L=1.2$时求β值和斜率$dz(0)/d\eta$。通过trapz完成积分。

5.31 一均匀梁在$\eta=0$处固定，在$\eta=1$处不固定，求位移$y(\eta)$。边界条件为$y(0)=dy(0)/d\eta=d^2y(1)/d\eta^2=d^3y(1)/d\eta^3=0$。从$\eta=0.5$到$\eta=1$处对梁施加一均匀的单位负载，画出梁的位移曲线图。

5.32 蜗轮的超前角为λ，比率$\beta=N_1/N_2$，其中N_1为蜗轮齿数，N_2为从动轮齿数，轴间中心距为C，法向径节P_{dn}为[②]：

$$K=\frac{2P_{dn}C}{N_2}=\frac{\beta}{\sin\lambda}+\frac{1}{\cos\lambda}$$

在实际中，$1\leqslant K\leqslant2$，$1°\leqslant\lambda\leqslant40°$，$0.02\leqslant\beta\leqslant0.30$。对特定值的组合，$\lambda$可有一个值、两个值或无值。

(a) 当$\beta=0.02,0.05,0.08,0.11,0.15,0.18,0.23$和$0.30$时，求使$K$最小的$\lambda$值。

(b) $K=1.5$，$\beta=0.16$时，求λ值。

5.33 习题1.14中，从压力为p_0，并处于可逆绝热条件下的容器中溢出气体的质量流量和下式成比例：

$$\psi=\sqrt{\frac{k}{k-1}}\sqrt{\left(\frac{p_e}{p_0}\right)^{2/k}-\left(\frac{p_e}{p_0}\right)^{(k+1)/k}}$$

其中，p_e是容器的外部压力，k是绝热可逆气体常数。极大值出现在：

$$\frac{p_e}{p_0}=\left(\frac{2}{k+1}\right)^{k/(k-1)}$$

$k=1.4$时，用fminbnd和min验证这一极大值，min利用$0\leqslant p_e/p_0\leqslant1$范围内的200个等间隔值进行计算。

5.34 (a) 用fsolve搜索以度数表示的θ值和k值，并使其在$a=1$，$b=3$时满足下列方程：

$$b=k(1-\cos\theta)$$

① 参见 M. Irvine，出处同上，第4页。

② 参见 M. F. Spotts 和 T. E. Shoup, *Design of Machine Elements*, Prentice Hall, Upper Saddle River, NJ, 1998年，第613页。

$$a = k(\theta - \sin\theta)$$

(b) (a)中的两个方程可合并为下面的一个方程：

$$b(\theta - \sin\theta) - a(1 - \cos\theta) = 0$$

当 $a = 1, b = 3$ 时，用fzero求 θ 值，然后用(a)中的一个方程求取 k 值。

[答案：$k = 6.9189, \theta = 55.4999°$。]

5.35 (a) $\sigma = 5.667 \times 10^{-8}$，$T_1 = 373$ K，$T_2 = 293$ K 时，用fsolve从以下方程求取 Q，T_A和 T_B值。

$$T_1^4 - T_A^4 = Q/\sigma$$
$$T_A^4 - T_B^4 = Q/\sigma$$
$$T_B^4 - T_2^4 = Q/\sigma$$

(b) (a)中的方程也可以写为：

$$\begin{bmatrix} 1 & 0 & 1/\sigma \\ 1 & -1 & -1/\sigma \\ 0 & 1 & -1/\sigma \end{bmatrix} \begin{Bmatrix} x \\ y \\ Q \end{Bmatrix} = \begin{Bmatrix} T_1^4 \\ 0 \\ T_2^4 \end{Bmatrix}$$

其中，$x = T_A^4$，$y = T_B^4$。使用左除从方程组中求取 Q，T_A和 T_B值。

[答案：$T_A = 352.052$，$T_B = 326.5116$，$Q = 226.4312$。]

5.36 使用符号工具箱求下列表达式的极限。

(a) $\lim\limits_{x\to 0}(1 - \sin(2x))^{1/x}$

(b) $\lim\limits_{x\to 1}\dfrac{\ln x^n}{1 - x^2}$

5.37 给定公式：

$$f(x) = \frac{1 - e^{-x}}{1 + x^3}$$

若 $x = a + b\cos(w)$，用符号工具箱中的函数求出在 $a = 1.2$，$b = -0.45$，$w = \pi/3$ 时 $f(x)$的值。

5.38 使用solve函数求出在 $0 \leq x \leq \pi$ 范围内使 $f(x)$取得极值的 x 的值，并指出是极大值还是极小值。

$$f(x) = e^{\sin x}$$

5.39 从 0 到 4π 之间对于 b 取 10 个值，求下式的积分结果：

$$A(b) = \int_0^b \frac{2x + 5}{x^2 + 4x + 5}dx$$

5.40 下式是两自由度系统中某一质量在特定数值集合下位移对无量纲时间 τ 的拉普拉斯变换①：

$$X_1(s) = \frac{0.1s^3 + 0.0282s^2 - 0.0427s + 0.0076}{s^4 + 0.282s^3 + 4.573s^2 + 0.4792s + 2.889}$$

画出在 $0 \leq \tau \leq 35$ 范围内的反拉普拉斯变换 $X_1(s)$。

① B. Balachandran 和 E. B. Magrab，出处同上，第 428 页。

第6章　二维图形

本章介绍了许多二维绘图功能的实现方法。

6.1　概述

MATLAB 提供了很多灵活易用的二维和三维绘图功能函数。这些绘图函数分为三类：图形处理、曲线和曲面图的生成，以及注释和图形特性。绘图函数虽然很多，但语法大致相同，可归为同类函数。本章和下一章用到的函数包括：

管理类	生成类	注释和特性类
figure	**2-D**	xlabel
subplot	plot	ylabel
zoom	polar	text
hold	fill	title
	plotyy	legend
3-D	semilogx, semilogy, loglog	box
view		set
rotate3d	stairs	grid
	stem	axis, axis equal, axis off
	bar	clabel
	3-D	**3-D**
	plot3	text3
	surf, surfc	zlabel
	mesh, meshc, meshz	colorbar
	contour, contour3, contourf	colormap
		shading
	waterfall	
	cylinder	

生成图形实体时，应突出重点，以满足解的客观性。通过使用坐标轴标注、图形标题、曲线标注(如有多条曲线)及重要数值标注方法使其清晰易懂、特点明显。而且在不影响效果的前提下，对强调功能的颜色、线型、符号和文本也应加以使用。

一组典型的生成图形表达式包括处理函数，后面是一个或多个图形生成函数，然后是注释函数，可能其后还有附加的管理函数。除了管理函数，其余函数可以采用任意的顺序。而且，注释函数和图形属性函数是可选的。MATLAB 在坐标轴上标出刻度及刻度值，即使多值输出也如此。只要能得到部分标注的图形，就证明函数语法的使用是正确的。

一些图形管理函数

调用图形的管理、生成、注释和属性函数时，图形将显示在图形窗口中，该窗口由 MATLAB 在运行时生成。当一段程序(函数)用到几个图形函数时，MATLAB 将生成一个新的图形窗口。然而在生成新的图形窗口之前，任何先前生成的图形窗口将被移走。为在其自身的图形窗口中显示每一幅新图形，必须使用函数：

```
figure(n)
```

其中，n 是整数。如果省略 figure 参数，MATLAB 给出下一个整数值。

也可以用：

```
subplot(i,j,k)
```

将几个独立生成的图形放在同一图形窗口中。前两个参将窗口分为不同的块(行和列)，第三个参数指出图形放在第几个块中，值取 1 表示左上角的块，值等于行数和列数之和时表示应放于右下角的块。随着数字的增大，显示的部分按从左到右，从上到下的顺序定位。在程序中任何一个出现在 figure/subplot 之后的注释函数和管理函数仅适用于 subplot 函数第 3 个参数指定的块中。每一块中，均可使用二维和三维图形生成函数。参考图 6.1 中使用 figure 和 subplot 的几个例子。由图中可以看出，如果仅需要一个图形窗口，则即使用到了 subplot 函数，figure 也可以省略。

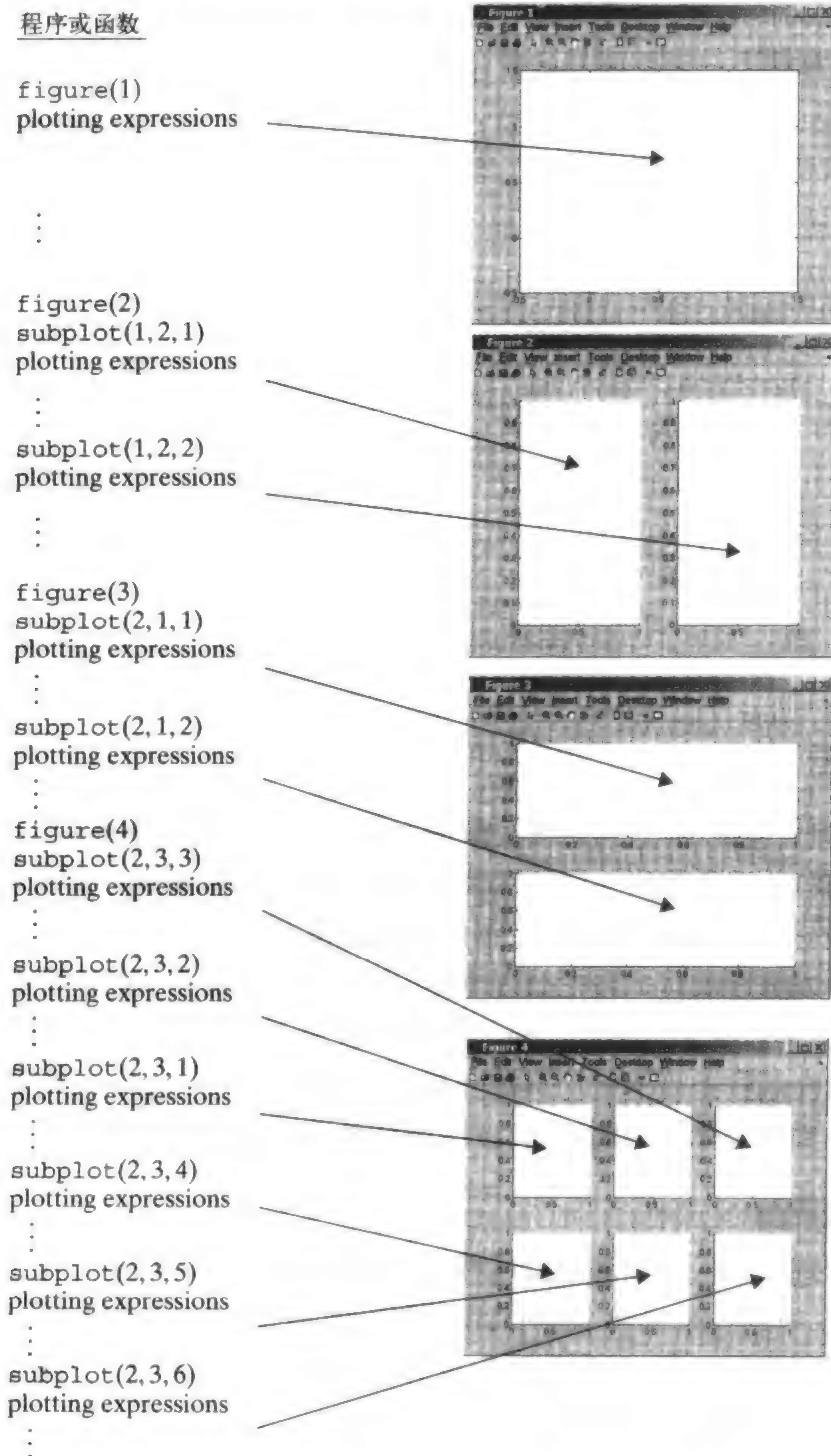


图 6.1　figure 和 subplot 不同组合应用示例

由于每一个图形生成函数创建一个新的图形窗口[①],因此,为了在一个图形中画出多条曲线、多个曲面图或线(或其组合),必须使用:

```
hold on
```

该语句保持当前窗口(或 subplot 子块)为激活状态。在每个图形窗口中,所有图形均可通过 *Edit* 下拉菜单 *Copy Figure* 命令复制到 Windows 剪切板。可将图形传输(粘贴)到字处理程序的页中并保存为 Windows 图元文件格式。

MATLAB 提供了将图形转换为与多种普通打印设备相兼容的格式的方法。例如,如果将当前活动图形窗口的图形存为二层压缩文件,名为 *FileName.eps*,然后在黑白打印机上输出,可使用以下语句:

```
print -deps2 'c:\path\FileName.eps'
```

或:

```
print('-deps2','c:\path\FileName.eps')
```

其中,关键字 - *deps*2 表示创建二层压缩文件,*path* 指明文件所在的目录和子目录。其他选项参见 print 的帮助文件。另一方面,如果要将二层压缩文件插入到 MS Word 文档,以显示图形的"tiff"预览图像,可以用以下语句[②]:

```
print -deps2 -tiff 'c:\path\FileName.eps'
```

或:

```
print('-deps2','-tiff;'C:\path\FileName.eps')
```

其中,关键字 - *tiff* 表示可获得图形的"tiff"预览图像。

6.2 基本二维绘图命令

基本二维绘图命令为:

```
plot(u1,v1,c1,u2,v2,c2,/..)
```

其中,u_j 和 v_j 分别是某点或一系列点的 x 和 y 坐标。或者是成对的数字、相同长度的向量、同阶次矩阵,或是计算时生成上述三者之一的表达式。c_j 是字符串:一个字符确定画线/点的颜色,另一个字符确定画点的类型,两个字符用于定义线的特征。表 6.1 给出了线和点的不同属性定义。要画一系列点时,c_j 可以是's'(用方框画点)或是'*'(用星号画点)。无论是否显示,都应用(直)线将这些点连接起来,字符 c_j 可以是'-',代表实线;也可以是'--',代表虚线。以相同的颜色绘制线和点时,c_j 包含两种描述符。例如,要用蓝色虚线连接蓝色菱形点时,c_j 为'b--d',单引号内三个字符的顺序不重要。当点和线一起画,而且定义线的点与要画的点数目

① MATLAB 窗口的外观、管理和文件管理的描述取决于 Windows 环境。其他操作系统的使用过程与之类似。

② 为在 MS Word 中使用该文件,必须安装相应的相缩过滤器。过滤器作为 MS Word 的一部分,却不是 MS Word 的默认安装。本例中,可运行 MS Word 的安装程序并安装相应的的过滤器。

不同时,c_1 定义线型符号,c_2 定义画点符号,反之亦然。如果省略 c_j,则使用系统默认值。如所画曲线多于一条,则曲线颜色按默认顺序变化。

下面给出的方法可以画出点、线、圆、表达式、曲线族和多个函数描述的曲线。

表 6.1　线和点的特性

线的类型		线的颜色		点的类型	
符号	说明	符号	说明	符号	说明
-	实线	r	红	+	加号
--	虚线	g	绿	o	圆
:	点线	b	蓝	*	星号
-.	点划线	c	青	.	点
		m	深红	x	十字线
		y	黄	s	方框
		k	黑	d	菱形
		w	白	∧	正三角
				∨	倒三角
				>	右三角
				<	左三角
				p	五角星
				h	六角星

6.2.1　画点

在(2,4)坐标处加一红色星号,语句为:

```
plot(2,4,'r*')
```

6.2.2　画线

用默认线型(实线)及默认颜色(蓝色)画一条从(0,0)到(1,2)的直线,语句为:

```
plot([0,1],[0,2])
```

第 1 个二元向量[0,1]为 x 坐标值,第 2 个二元向量[0,2]为 y 坐标值。因此,每一向量的第 1 个元素定义了线段起点的(x,y)坐标,第 2 个元素定义了终点的坐标。

设想画 n 条互不相连的线段,每条线段的两个端点坐标分别是(x_{1n},y_{1n})和(x_{2n},y_{2n})。为此,创建 4 个向量:

$$\boldsymbol{x}_j=[x_{j1},x_{j2}\cdots x_{jn}]$$
$$\boldsymbol{y}_j=[y_{j1},y_{j2}\cdots y_{jn}]\qquad j=1,2$$

则 plot 指令为:

```
plot([x1;x2],[y1;y2])
```

其中,[x1;x2]和[y1;y2]都是$(2\times n)$阶矩阵。

为说明这个表达式,在 $x=2,4,6$ 和 8 处画 4 条从 $y=0$ 到 $y=\cos(\pi x/20)$的垂直线。程序如下:

```
x=2:2:8;
y=[zeros(1,length(x));cos(pi*x/20)];
plot([x;x],y,'k')
```

由于 $x_1=x_2=x$,因此所有线段都具有相同的颜色(本例中为黑色)。函数 zeros 用于创建与 x 等长的 0 向量。结果如图 6.2(a)所示。但是,MATLAB 会自动标出轴线刻度,使得第一条和最后一条直线与图形的边轴重合,因而变得不可见。为此,可使用以下语句对轴进行调整以使直线可见。

```
axis([xmin,xmax,ymin,ymax])
```

其中,x_{min},x_{max},y_{min}和 y_{max}分别是 x 轴和 y 轴的最小值和最大值。修改后的程序为:

```
x=2:2:8;
y=[zeros(1,length(x);cos(pi*x/20)];
plot([x;x],y,'k')
axis([1,9,0,1])
```

修正图形如图 6.2(b)所示。

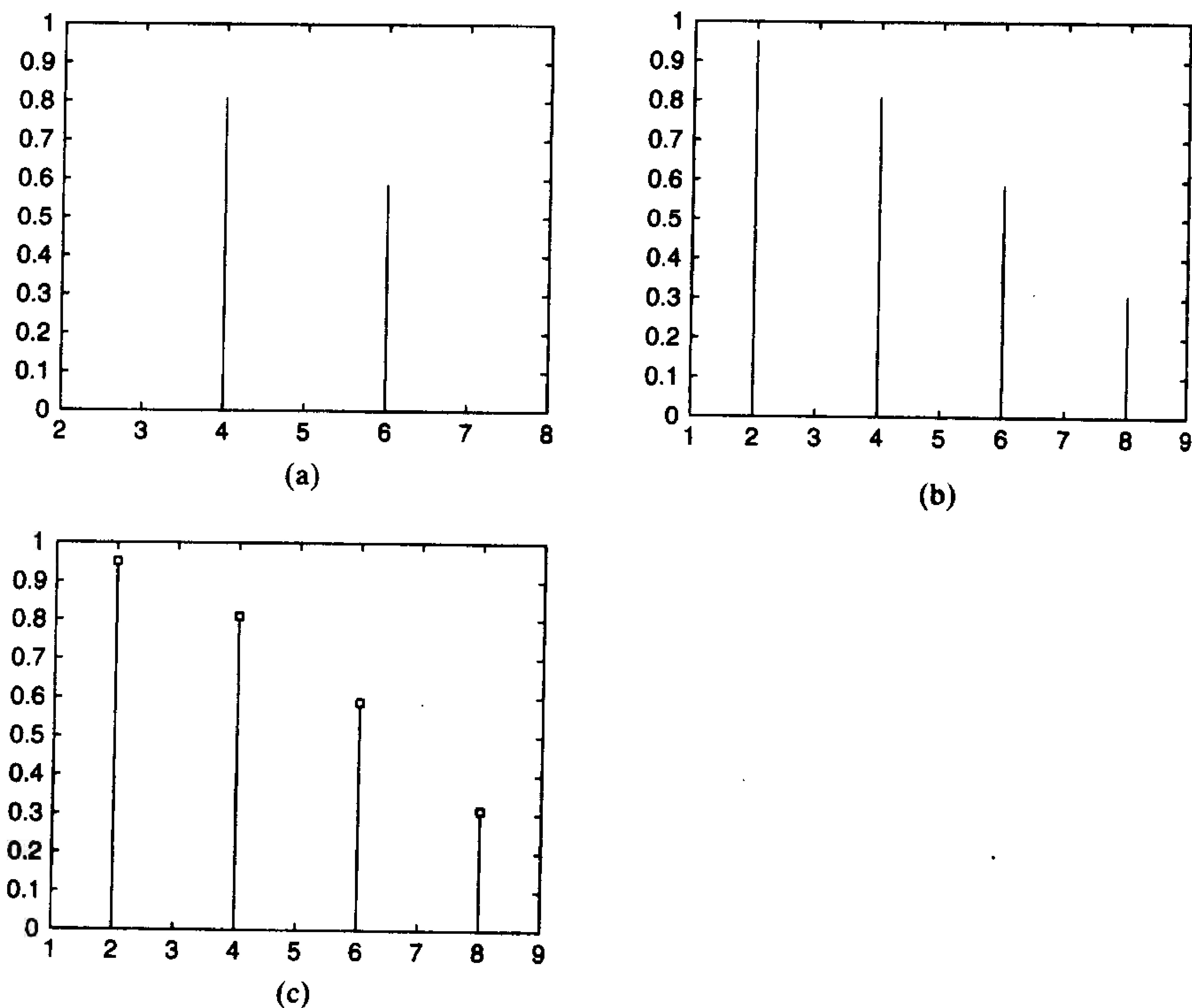


图 6.2　(a) 图框覆盖了线段;(b) 使用 axis 扩充坐标轴的范围,可以看到全部线段;(c) 在线段顶端放置红色方框

得到轴的边界值,并根据情况对其进行重新定义可以提供更大的灵活性。边界值可由以下语句获得:

```
v = axis;
```

其中,v 是一个四元向量:

$$v(1) = x_{min} \qquad v(3) = y_{min}$$
$$v(2) = x_{max} \qquad v(4) = y_{max}$$

因此,为获得如图 6.2(b)所示的修正图形,可进一步修改程序如下:

```
x = 2:2:8;
y = [zeros(1,length(x));cos(pi * x/20)];
plot([x;x],y,'k')
v = axis;
v(1) = 1;
v(2) = 9;
axis(v)
```

如果希望在每条垂直线的顶部放置红色方框,则需要在 plot 函数中加入三组参量,程序为[①]:

```
x = 2:2:8;
y = [zeros(1,length(x));cos(pi * x/20)];
plot([x;x],y,'k',x,cos(pi * x/20),'rs')
axis([1,9,0,1])
```

结果如图 6.2(c)所示。

6.2.3 画圆

在笛卡儿坐标系中画一个半径为 r,圆心坐标为(a,b)的圆,首先应进行以下坐标变换(参见图 2.2)。

$$x = a + r\cos(\theta)$$
$$y = b + r\sin(\theta)$$

其中,$0\leqslant\theta\leqslant\theta_1\leqslant 2\pi$。当 $\theta_1 < 2\pi$ 时,图形为一段圆弧。设 $\theta_1 = 2\pi, a = 1, b = 2, r = 0.5$,则画圆的程序如下:

```
theta = linspace(0,2 * pi);
plot(1 + 0.5 * cos(theta),2 + 0.5 * sin(theta))
axis equal
```

函数 axis equal 使图形对称,因此图形是圆形,而不是椭圆。执行结果如图 6.3 所示。

画出 6 个同心圆,其初始半径为 0.5,半径增量为 0.25,圆心处标上加号,程序如下:

```
theta = linspace(0,2 * pi,50);          %(1 × 50)
rad = 0.5:0.25:1.75;                    %(1 × 6)
x = 1 + cos(theta)' * rad;              %(50 × 6)
```

① 从某方面来看,认为 plot 表达式是图形函数 stem 的范化。而 stem 假定 $x_1 = x_2, y_1 = 0$。

```
y = 2 + sin(theta)' * rad;            %(50×6)
plot(x,y,'k',1,2,'k+')
axis equal
```

按列画出矩阵中的值。因为对应每个 *rad*(弧度)值要画出 50 个 θ 值,所以定义矩阵阶次为(50×6)。如果省略字符串'k',则每个圆的颜色不同。程序执行后生成图 6.4。

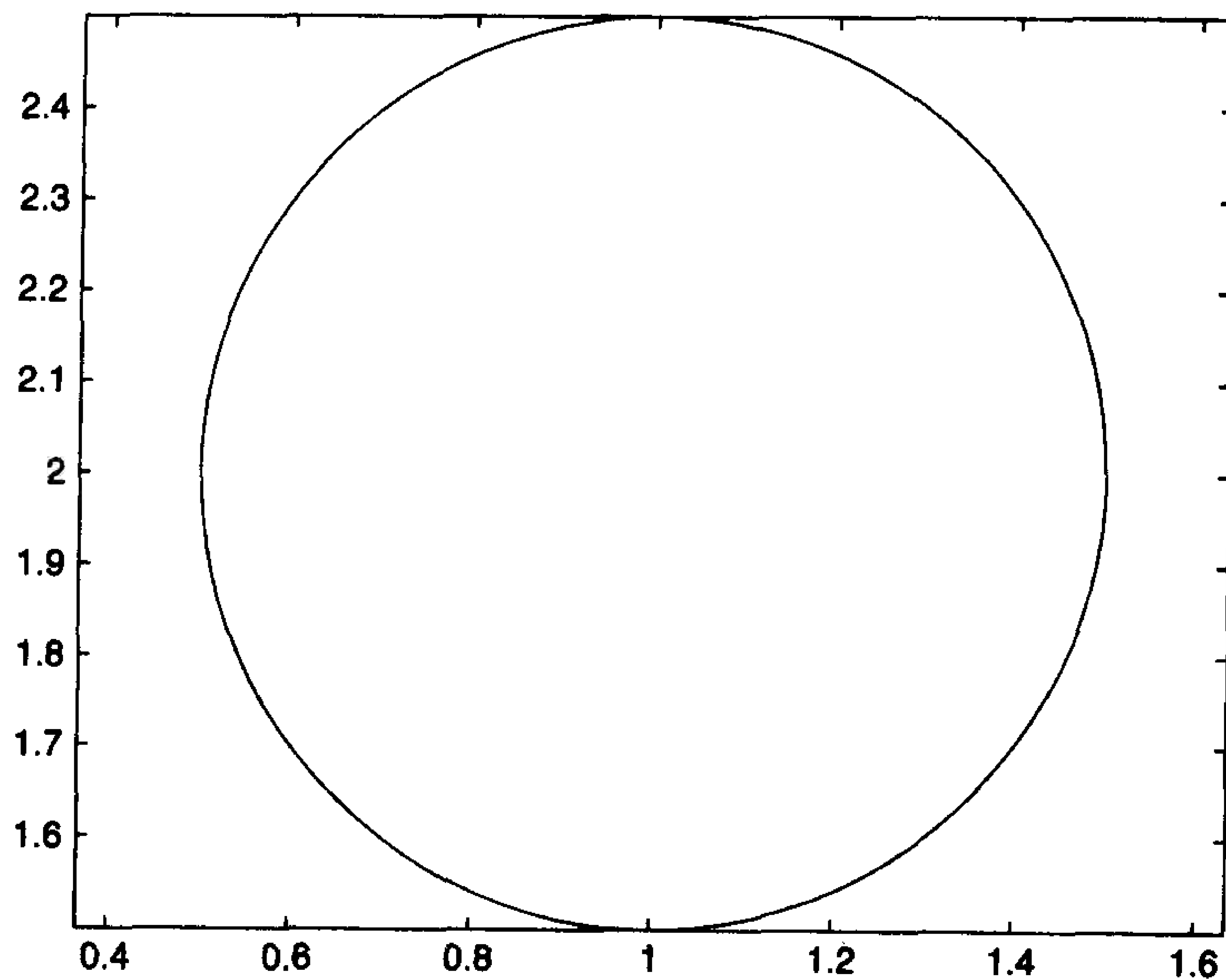


图 6.3　圆

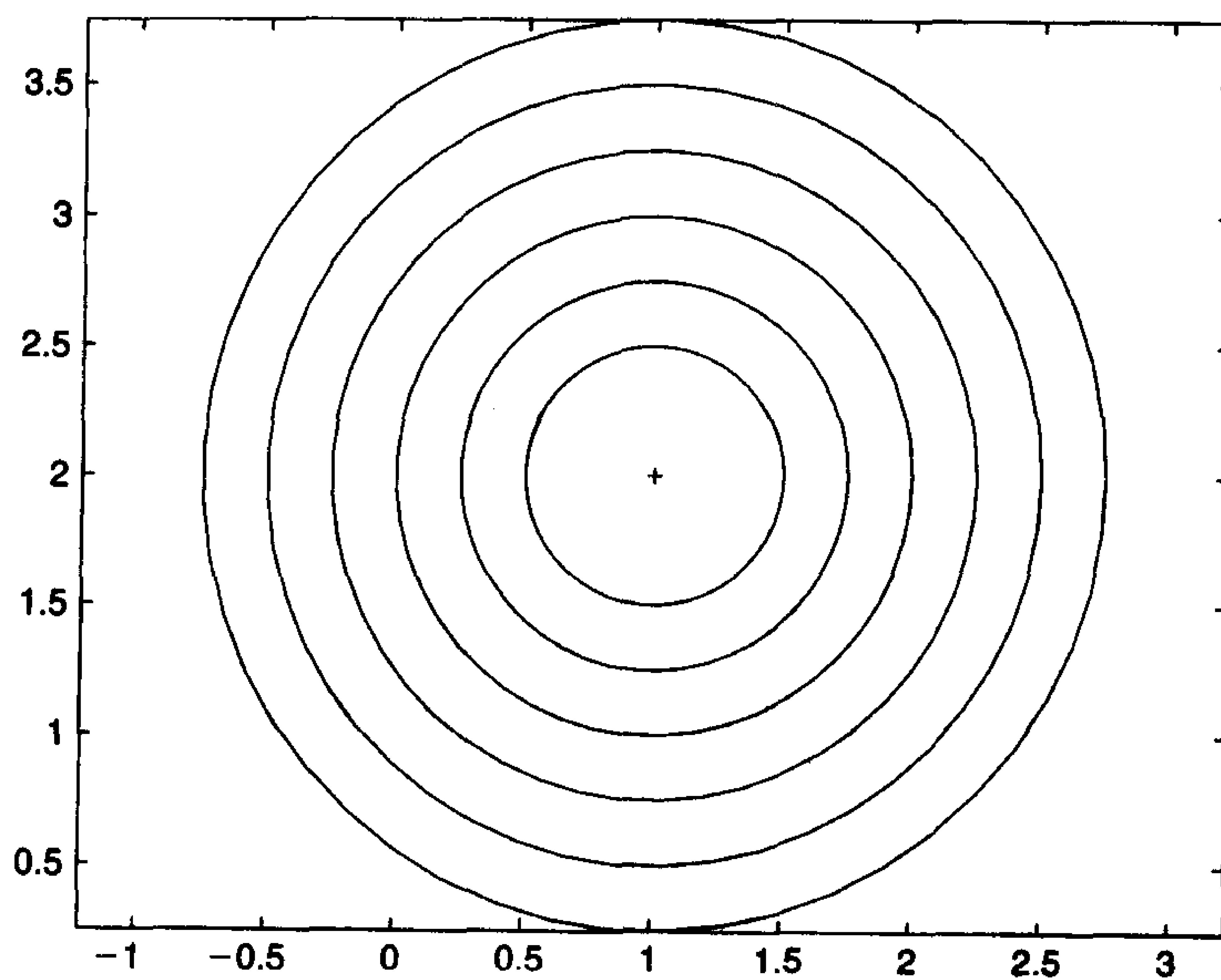


图 6.4　6 个同心圆

6.2.4 曲线族的绘制

前面绘制 6 个同心圆的例子中给出了一种绘制曲线族的方法。一般来说,MATLAB 允许用向量表示一个轴,用矩阵表示另一个轴。并且,根据矩阵的行和列与向量长度的匹配情况,画出向量对匹配的行或列的曲线图。

画抛物线族的情况如下:

$$y = a^2 - x^2$$

其中,$-5 \leqslant x \leqslant 5$,$a = 1,2,\cdots,5$,程序如下:

```
x= -5:0.2:5;
a=1:5;
[xx,aa]=meshgrid(x.^2,a.^2);
plot(x,aa-xx,'k')
```

结果如图 6.5 所示。

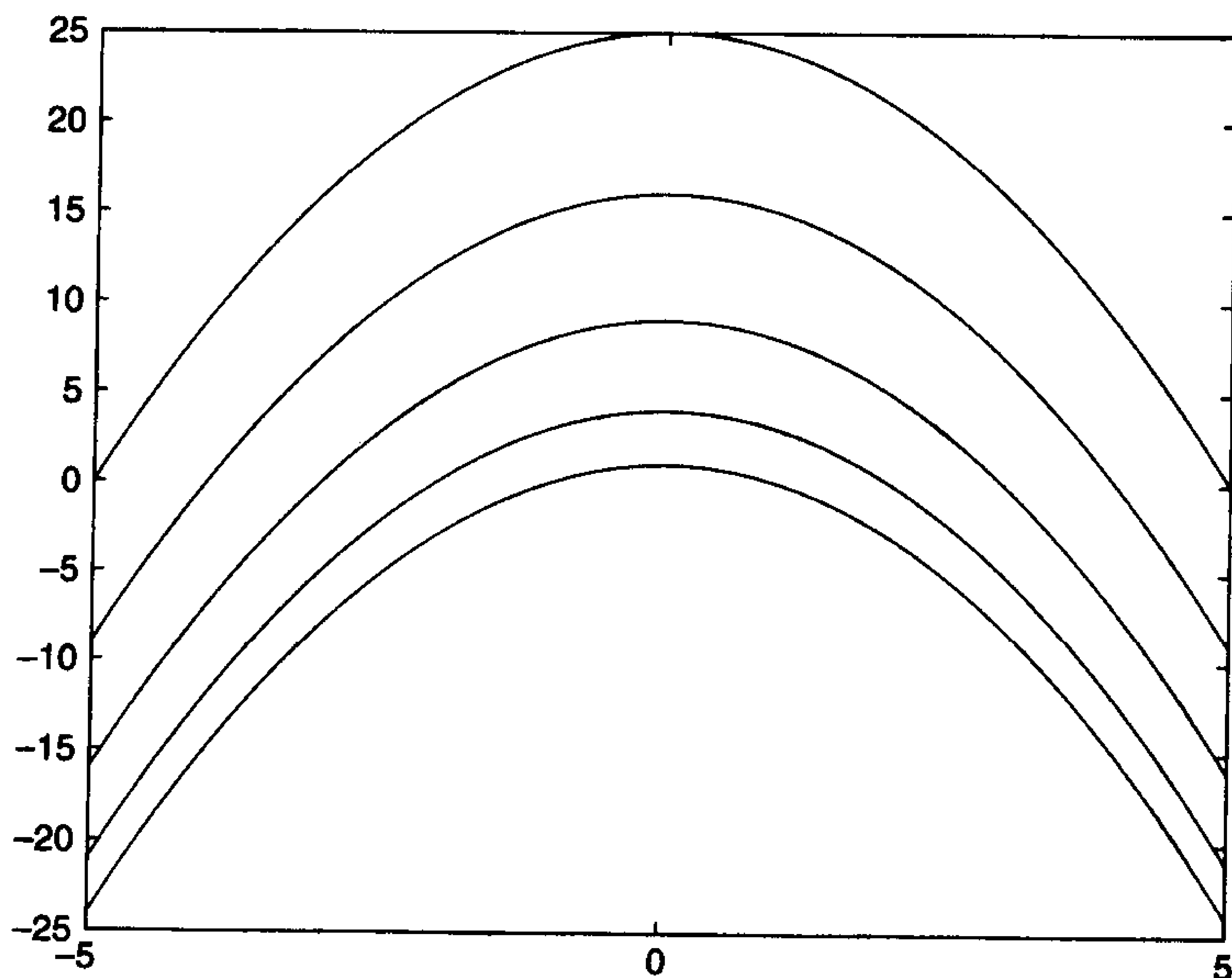


图 6.5 抛物线族

考虑级数收敛的图形表示:

$$S_N = \sum_{j=1}^{N} \frac{1}{(a+j)^2}$$

其中,$N = 1,2,\cdots,10$ 并且 $a = 1,2,3$。该例中,应用 cumsum 函数(参见 2.5 节),程序如下:

```
aa=1:3;                      %(1×3)
N=1:10;                      %(1×10)
[a,k]=meshgrid(aa,N);        %(10×3)
S=cumsum(1./(a+k).^2);       %(10×3)
plot(N,S,'ks-')
```

结果如图 6.6 所示。

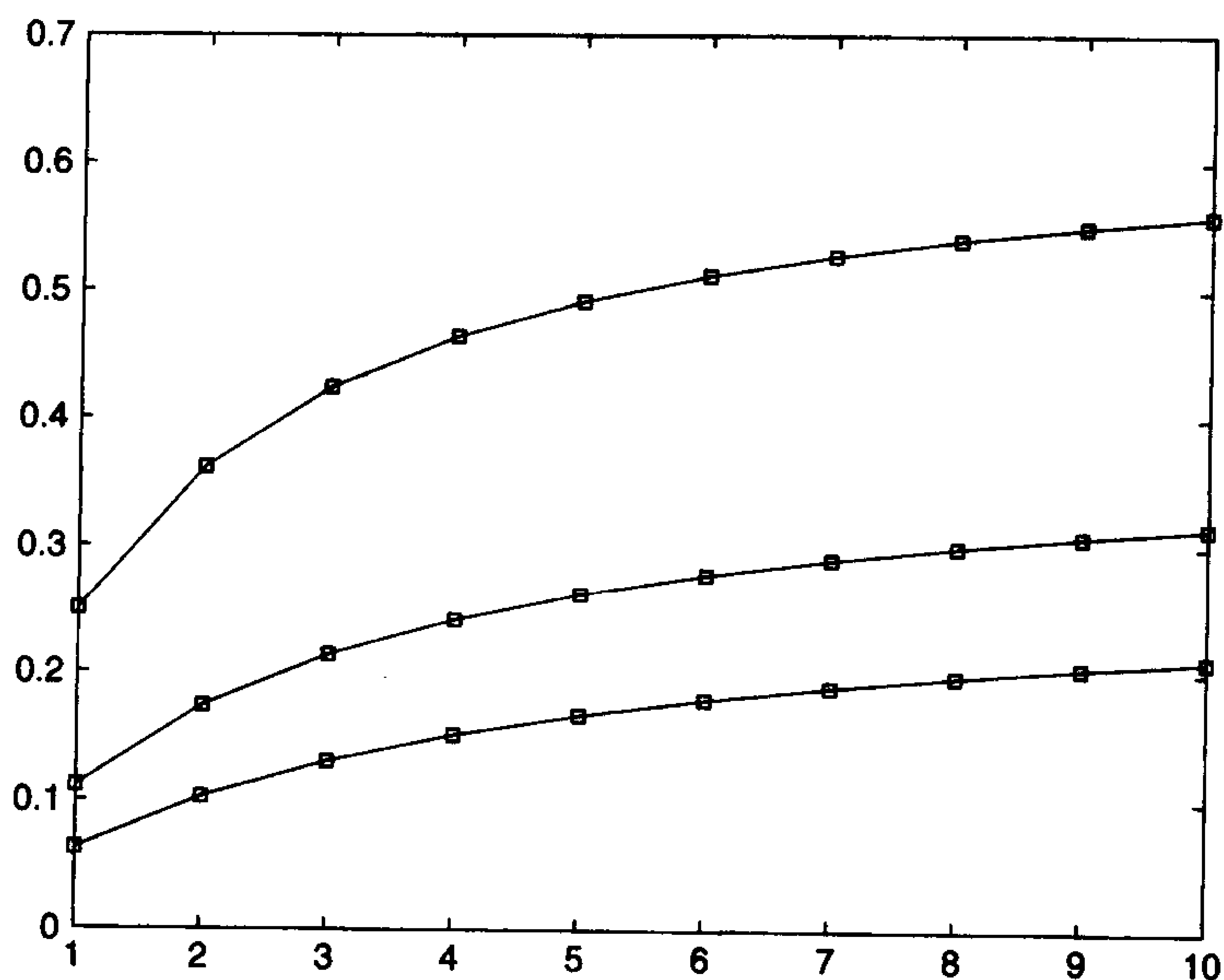


图 6.6　级数收敛的图形表示

6.2.5　在一个图形界面中绘制多个函数图形①

考虑以下三个函数：

$$g_1(x)=0.1x^2$$
$$g_2(y)=\cos^2 y$$
$$g_3(z)=e^{-0.3z}$$

其中，$0\leqslant x=y=z\leqslant 3.5$。可用以下三种方法中的任何一种画出这三条函数曲线：

```
x = linspace(0,3.5);
plot(x,[0.1 * x.^2;cos(x).^2;exp( - 0.3 * x)],'k')
```

或

```
x = linspace(0,3.5);
plot(x,0.1 * x.^2,'k',x,cos(x).^2,'k',x,exp( - 0.3 * x),'k')
```

或

```
x = linspace(0,3.5);
plot(x,0.1 * x.^2,'k')
hold on
plot(x,cos(x).^2,'k')
plot(x,exp( - 0.3 * x),'k')
```

执行任一程序均可生成图 6.7(a)，其中的所有曲线均为黑色。

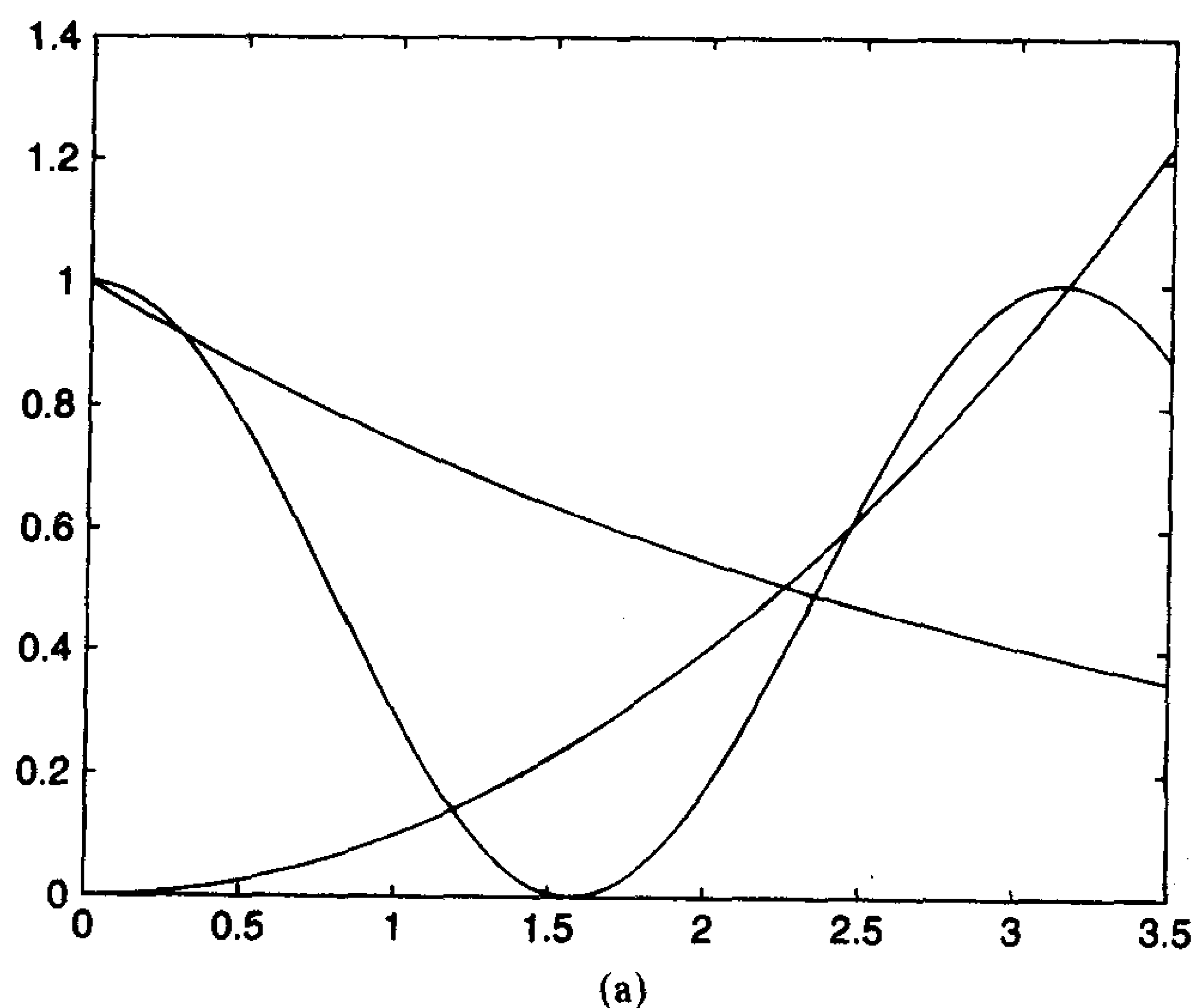
另一方面，如果这些函数中自变量的范围不同，则只有第 2 段和第 3 段程序可用。例如，

① 用 plotyy 在两种不同纵坐标内绘制两种不同类型的图形(参见 6.2.7 节的图 6.8)。

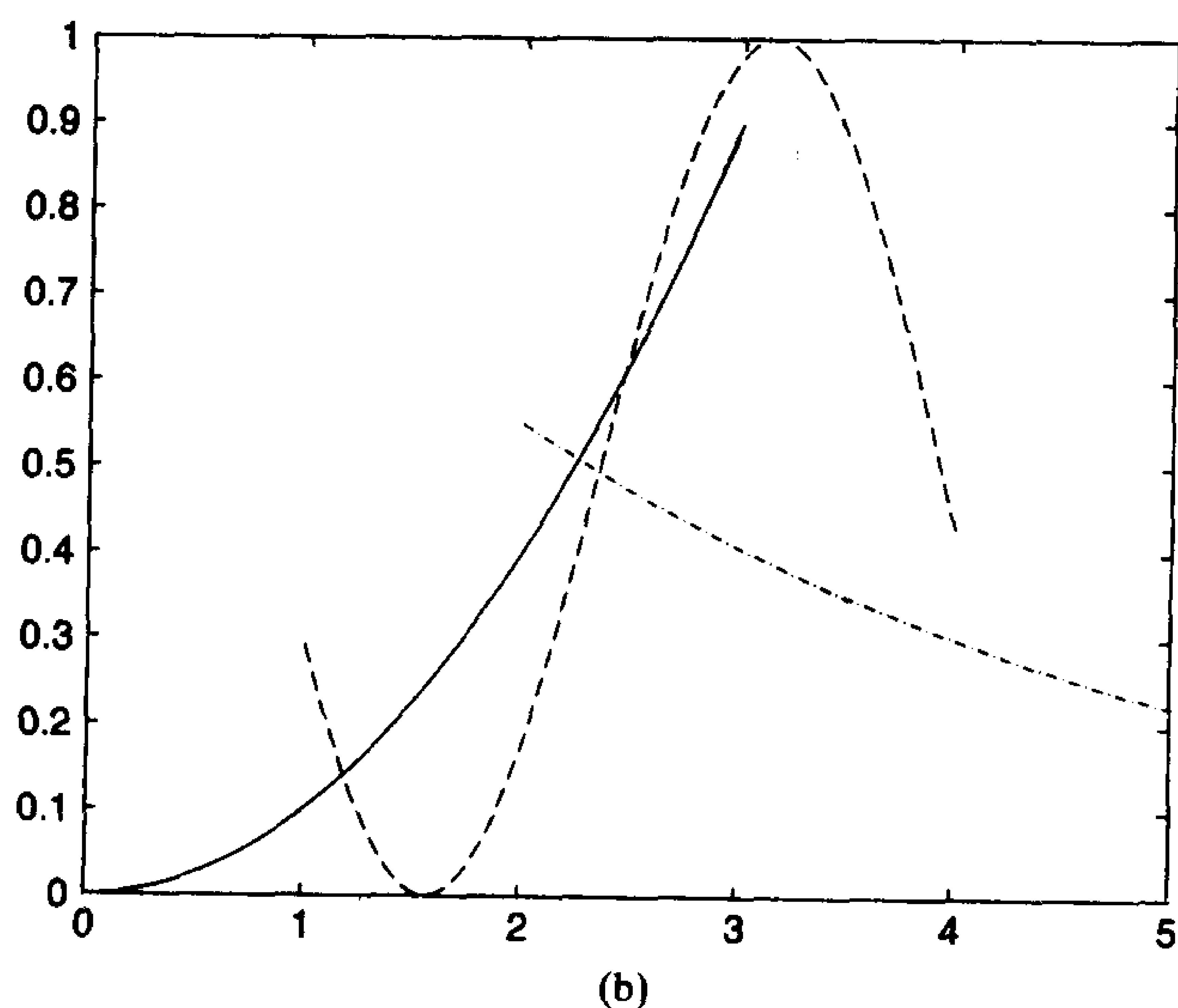
如果 $0\leqslant x\leqslant 3,1\leqslant y\leqslant 4,2\leqslant z\leqslant 5$,则第 2 段程序使用如下：

```
x = linspace(0,3,45);
y = linspace(1,4,55);
z = linspace(2,5,65);
plot(x,0.1 * x.^2,'k - ',y,cos(y).^2,'k - - ',z,exp( - 0.3 * z),'k - .')
```

结果如图 6.7(b)所示。注意,绘制每个函数图形时使用了不同线型,且绘制每条曲线所使用的点数不同。



(a)



(b)

图 6.7　(a) 同一范围内的 3 个不同函数图形；
(b) 不同范围内的3个不同函数图形

6.2.6 改变图形外观

以下几个函数可用于改变图形外观：

axis on 或 axis off [默认情况下为 on]
box on 或 box off [默认情况下为 on]
grid on 或 grid off [默认情况下为 off]

函数 box on 只有在函数 axis on 应用的情况下方可应用。

下面通过绘制利萨如(Lissajous)图形说明上述函数如何改变图形的外观。在坐标图上画出 $\sin(n\theta)$ 对 $\sin(m\theta+\theta_1)$ 的关系图就得到利萨如图形，其中 m，n 是正数，$0\leqslant\theta\leqslant 2\pi$，$0\leqslant\theta_1<2\pi$。设 $n=1$，$m=2$，$\theta_1=\pi/4(45°)$。如果将 θ 进行 101 等分，则程序如下：

```
th = linspace(0,2 * pi,101);
plot(sin(th),sin(2 * th + pi/4))
```

程序执行结果及用函数 box，axis 和 grid 修正后的结果参见表 6.2。

表 6.2 函数 box，grid 和 axis 的说明

函数	程序	图 形

box on
grid on
	th = linspace(0,2 * pi,101);
	x = sin(th);
	y = sin(2 * th + pi/4);
	plot(x,y,'k - ')
	box on
	grid on	

box off
grid off
axis off
	th = linspace(0,2 * pi,101);
	x = sin(th);
	y = sin(2 * th + pi/4);
	plot(x,y,'k - ')
	box off
	grid off
	axis off	

(续表)

函数	程序	图　形

box off
grid off
axis on
	th = linspace(0,2 * pi,101);
	x = sin(th);
	y = sin(2 * th + pi/4);
	plot(x,y,'k - ')
	box off
	grid off	

6.2.7 特殊功能图形

MATLAB 提供了很多可广泛应用的特殊功能图形函数。其应用情况可通过绘制下列表达式的不同图形加以说明：

$$F(\Omega) = H(\Omega)e^{j\theta(\Omega)} \quad \Omega \geqslant 0$$

其中，

$$H(\Omega) = \frac{1}{\sqrt{(1-\Omega^2)^2 + (2\zeta\Omega)^2}}$$

$$\theta(\Omega) = \tan^{-1}\frac{2\zeta\Omega}{1-\Omega^2}$$

且 $\zeta < 1$。

首先，创建函数 **FOm** 代替表达式：

```
function[H, T] = FOm(Om, z)
T = atan2(2 * z * Om, 1-Om.^2) * 180/pi;
H = 1./sqrt(1-Om.^2).^2 + (2 * z * Om).^2);
```

其中，$T = \theta(\Omega)$ 单位为度，$z = \zeta$。因为分母存在符号变化，故应用了反正切函数的双参数形式。

第一组要考虑的特殊功能图形函数为 semilogx，semilogy 和 loglog。函数 semilogx 在 x 轴为以 10 为底的对数条件下绘制图形，函数 semilogy 在 y 轴以 10 为底的对数条件下绘制图形，而函数 loglog 则在两个坐标轴都为以 10 为底的对数条件下绘制图形。表 6.3 给出了上述 3 个图形函数的应用情况。

考虑图形函数 stairs，stem 和 bar。图形函数 stairs 以梯形线绘制各数据点，函数 stem 通过绘制从 x 轴到离散数据点的直线来绘制图形，而函数 bar 则通过绘制从 x 轴到数据点的填充矩形来绘制图形。表 6.4 给出了上述 3 个图形函数的应用情况。

表6.3 函数 semilogx,semilogy 和 loglog 的说明

绘图函数	程序	图 形
semilogx	Om = linspace(0.01,10,200); [H,T] = **FOm**(Om,0.05); stairs(Om,H)	
semilogy	Om = linspace(0.01,10,200); [H,T] = **FOm**(Om,0.05); semilogy(Om,H)	
loglog	Om = linspace(0.01,10,200); [H,T] = **FOm**(Om,0.05); loglog(Om,H)	

表 6.4 函数 stairs,stem 和 bar 的说明

绘图函数	程序	图形
stairs	Om = linspace(0.01,2,30); [H,T] = **FOm**(Om,0.05); stairs(Om,H)	
stem	Om = linspace(0.01,2,30); [H,T] = **FOm**(Om,0.05); stem(Om,H)	
bar	Om = linspace(0.01,2,30); [H,T] = **FOm**(Om,0.05); bar(Om,H,0.6)	

最后要讨论的特殊功能图形函数为 plotyy,该函数可实现在两个不同取值范围的坐标系内绘制两个不同的函数图形。为说明该函数的应用情况,在同一图形内绘制 $H(\Omega)$和$\theta(\Omega)$图形。程序如下:

```
Om = linspace(0,2,100);
[H,T] = FOm(Om,0.05);
plotyy(Om,H,Om,T)
```

执行结果如图 6.8 所示。

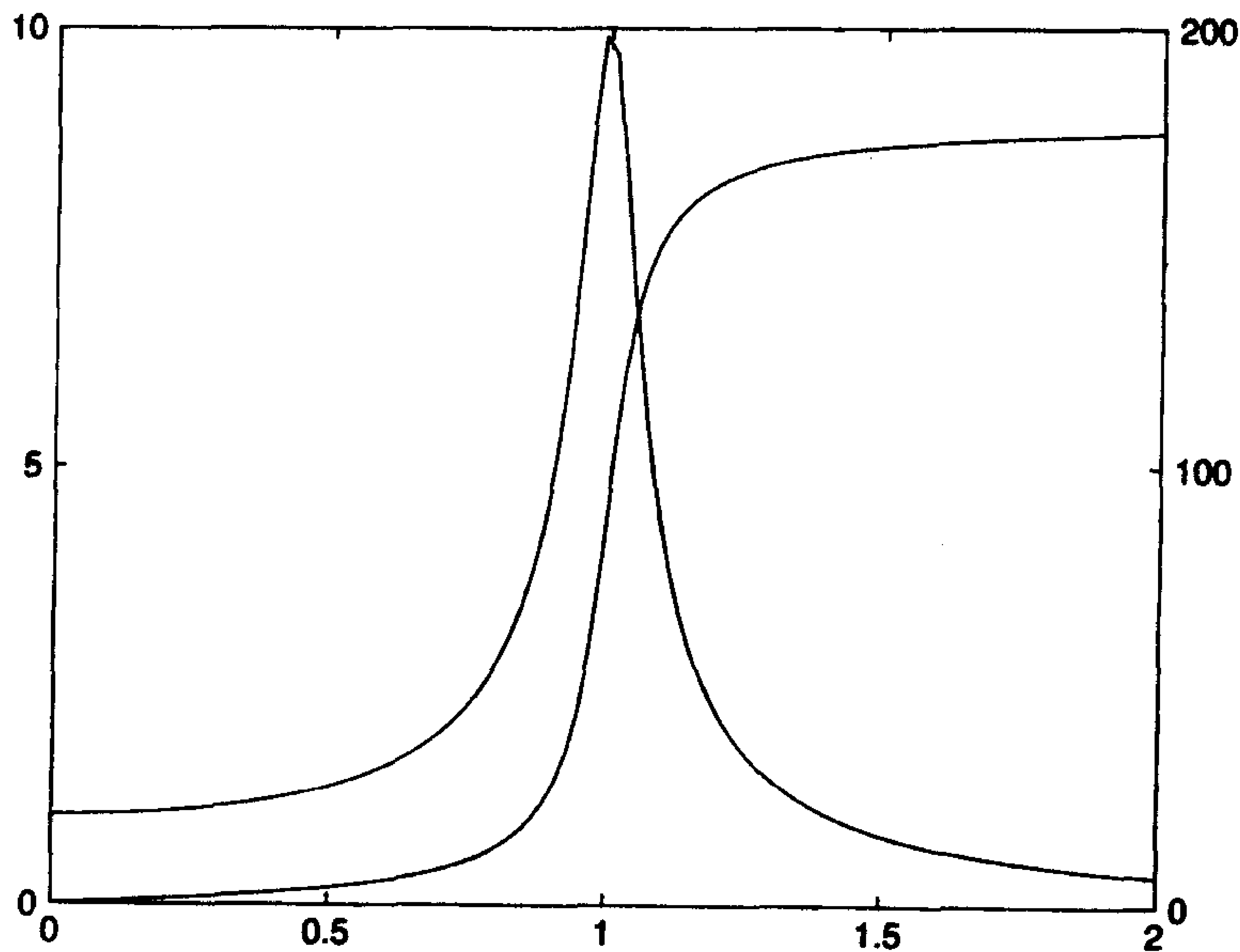


图 6.8　使用 plotyy 绘制 $H(\Omega)$和 $\theta(\Omega)$的图形

6.3　图形注释和可视化提高

6.3.1　概述

MATLAB 具有增强图形可视化功能。本节将举例说明如何提高图形的可视化性能。

- 使用轴标注、图形标题、曲线标注、图注、填充区域和添加文本
- 改变轴、曲线和文本属性
- 使用希腊字母、数学符号和上下标

6.3.2　坐标轴和曲线的标注、图形标题、图注和文本

用于标注 x 轴、y 轴及添加图形标题的函数分别为:

```
xlabel(s)
ylable(s)
titile(s)
```

其中,s 为字符串。在图形窗口中任意位置放置文本的函数为:

```
text(x,y,s)
```

其中,x 和 y 为放置字符串 s 的位置的坐标。

画出标注、标题并说明两条相交曲线 cos(x)和 1/cosh(x)的关系,其中 $0 \leqslant x \leqslant 6$。在该范围内,两条曲线在 $x = 4.73$ 处相交(参考例 5.8)。再画一条过交点的垂直线,并在相交点附近标出 x 值。程序如下:

```
x = linspace(0,6,100);
plot(x,cos(x),'k',x,1./cosh(x),'k',[4.73,4.73],[-1,1],'k')
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersectings curves')
text(4.8,-0.1,'x=4.73')
text(2.1,0.3,'1/cosh(x)')
text(1.2,-0.4,'cos(x)')
```

结果如图 6.9 所示。plot 函数执行后即已确定出标注文本所在的坐标值。也就是说,仅在写出程序前两句并给出结果图形之后即可确定标注文本的坐标值。然后是 text 函数语句。

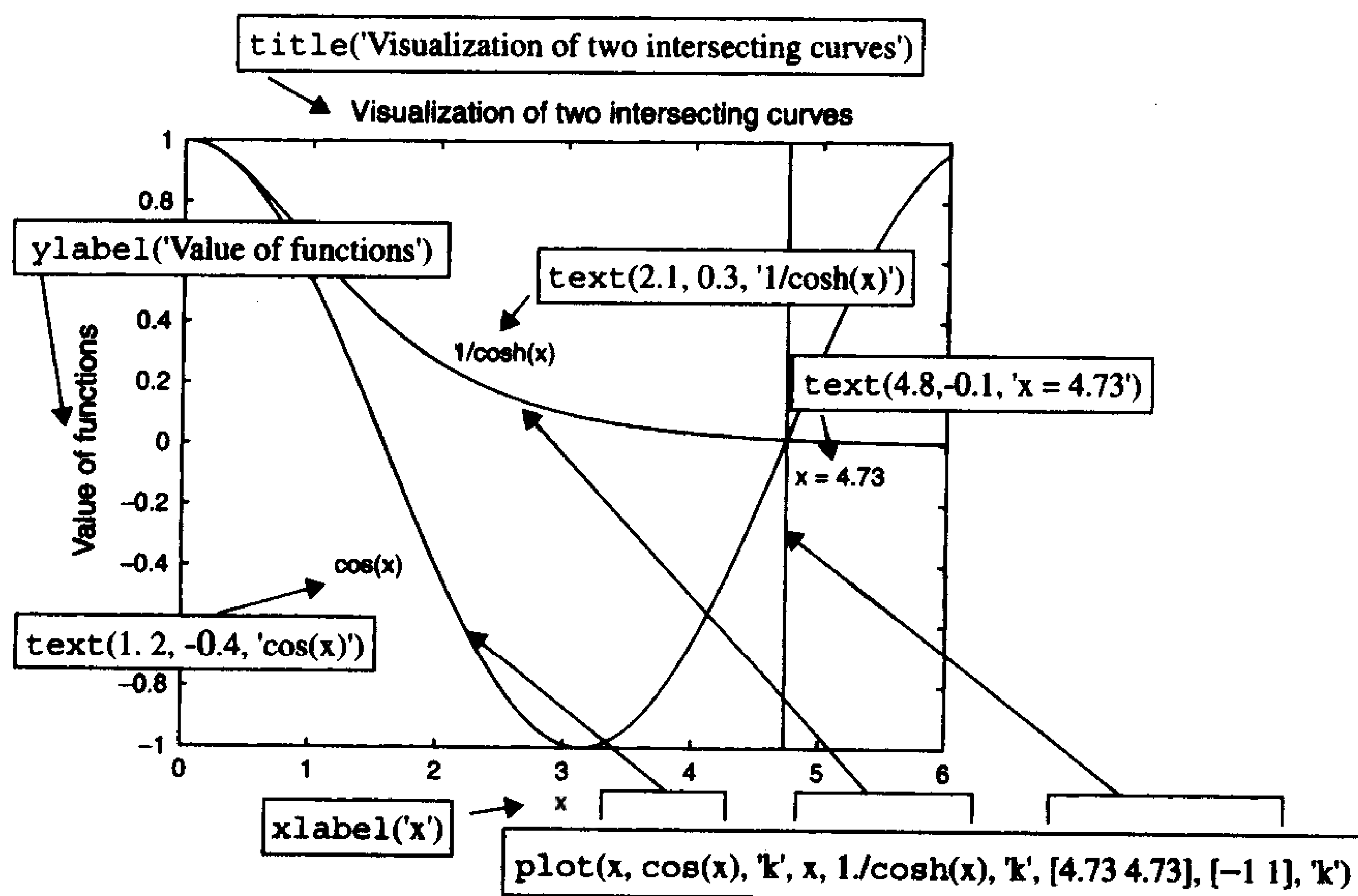


图 6.9　创建及标注所示图形的表达式

还有另一种标注图 6.9 所示曲线的方法,即使用:

```
legend(s1,s2,..., sn, 'Location',p)
```

其中,$s1$,$s2$ 等为每条曲线包含字符数字的标识符,按给出的顺序显示在图注框中。*Location* 为关键字,表明其后的字符串将给出图注框所处的位置。p 为字符串关键字,表明在图形内部或外部预定义的 8 个图注位置中图注所处的位置。legend 函数的参数个数与一个或多个 plot 函数绘制的不同线段个数相等,字符串顺序与曲线绘制顺序相同。因此,legend 函数放置在 plot 函数之后执行。当图注放置在图形坐标轴内部时,字符串 p 具有以下数值,当图纸放置在图形坐标轴外部时,参见函数 legend 的帮助文件。当图注放置在中间位置时,有:(1)图形上部,p = '*North*';(2)图形底部,p = '*South*';(3)图形右侧,p = '*East*';(4)图形左侧,p = '*West*'。将图注放置在其他位置时,有:(1)右上角,p = '*NorthEast*',这是字符串 p 的默认值;(2)左上角,p = '*NorthWest*';(3)右下角,p = '*SouthEast*';(4)左下角,p = '*SouthWest*'。当省

略这两个字符串参数时，图注放置在默认位置，即右上角。也可以用鼠标来放置图注。其方法是用鼠标点中图注，按下鼠标键将其拖至目标处。但这种放置仅是临时的，当图形关闭后下次重建时，图注仍显示在默认位置。legend 函数不同于 text 函数，因为 text 函数可多次使用，而 legend 却只可用一次。此外，legend 函数将所有图注放置在一个矩形框内。

为说明 legend 函数的应用，修改生成图 6.9 的程序，并用 legend 函数替换两个 text 函数。要求将图注放置在图形的左下角，即令 p = '*SouthWest*'。则程序为：

```
x = linspace(0,6,100);
plot(x,cos(x),'k-',x,1./cosh(x),'k--',[4.73, 4.73],[-1, 1],'k')
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8, -0.1,'x = 4.73')
legend('cos(x)','1/cosh(x)','Location','SouthWest')
```

程序执行结果如图 6.10 所示。注意，plot 函数绘制了 3 条曲线，但 legend 函数却只有两个字符串标识符，因此，只有前两条曲线进行了标注。plot 函数中每 3 个参数为一组，每组中的第 3 个参数表明用黑色绘制曲线，且 cos(x)显示为实线，1/cosh(x)显示为虚线。legend 函数参数与次序相关，第 1 个参数与第 1 条曲线对应，第 2 个参数与第 2 条曲线对应。如果使用了几个 plot 函数，则第 2 个 plot 函数的第 1 个参数接续上一个 plot 函数最后一个参数所对应的字符串标识符。每个 figure 或 subplot 中仅可使用一个 legend 函数。legend 函数中的位置关键字"*SouthWest*"表示将图注放置在图形的左下角。

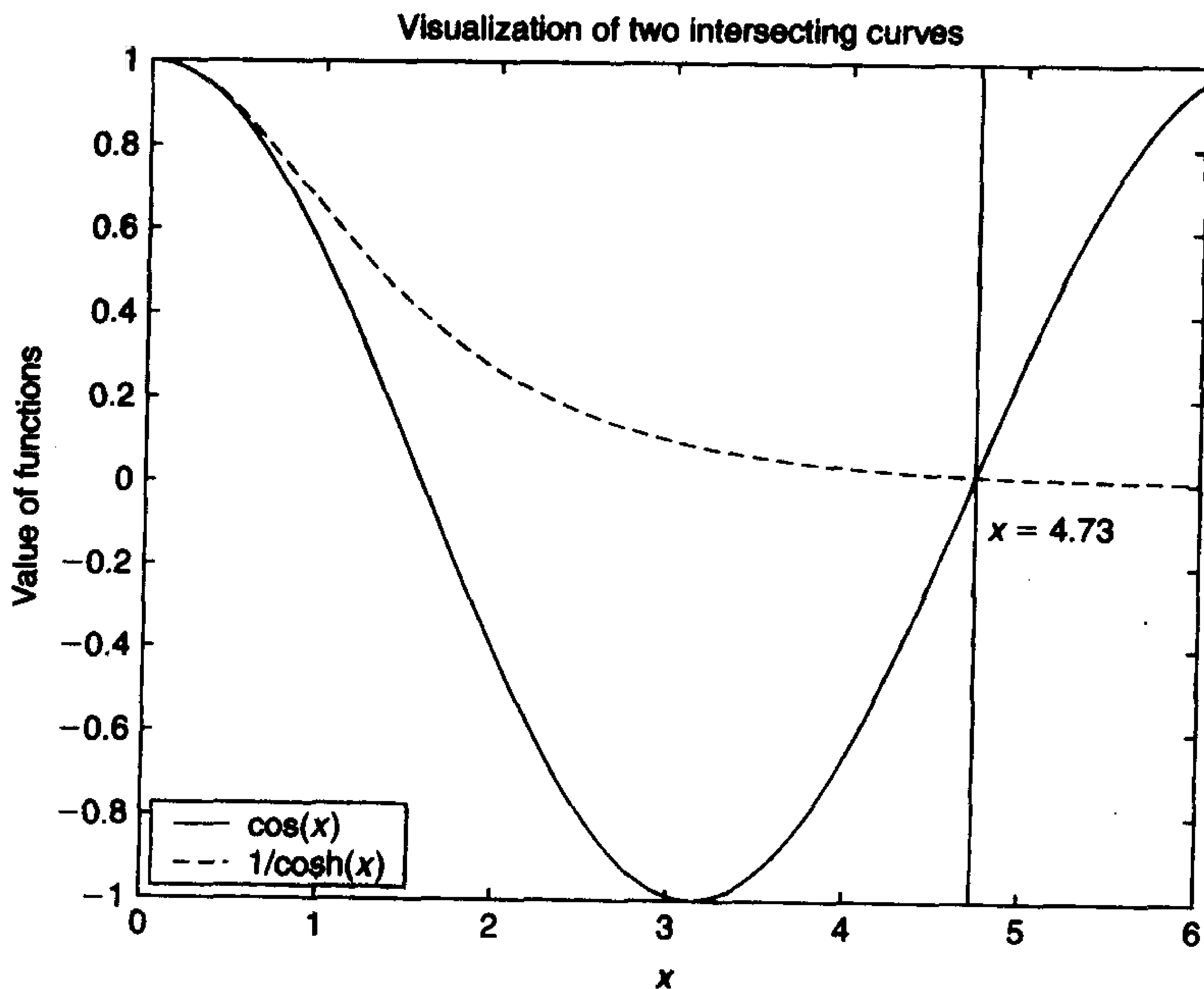


图 6.10　函数 legend 的使用

图 6.10 也可由以下程序获得：

```
x = linspace(0,6,100);
plot(x,cos(x),'k-')
hold on
plot(x,1./cosh(x),'k--')
plot([4.73,4.73],[-1,1],'k')
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-0.1,'x=4.73')
legend('cos(x)','1/cosh(x)','Location','South West')
```

如果未在 plot 函数中指定线型和颜色,则 legend 函数使用默认线型(实线)和默认颜色序列,即图注中的线段为不同颜色的实线。图注周围边框的显示或关闭可由下列函数控制:

```
legend('boxon')
```

和

```
legend('boxoff')
```

6.3.3 填充区域

图形区域可通过填充颜色或采用几何元素进行高亮处理,下面介绍几种高亮处理的方法。

填充多边形区域采用以下语句:

```
fill(x,y,c)
```

其中,x 和 y 是表示构成封闭多边形线段端点的同阶次矩阵,字符串 c 为表 6.1 中第 2 列给出的表示填充颜色的字符。

为说明 fill 函数的应用情况,可以修改以上程序,使两条曲线在 $0 \leqslant x \leqslant 4.73$ 范围内的区域填充为蓝绿色。多边形是由曲线 $1/\cosh(x)$ 在 $0 \leqslant x \leqslant 4.73$ 范围内的近似直线和曲线 $\cos(x)$ 在 $4.73 \geqslant x \geqslant 0$ 范围内的近似直线构成的。则程序修正为:

```
x = linspace(0,6,100);
plot(x,cos(x),'k-',x,1./cosh(x),'k--',[4.73,4.73],[-1,1],'k')
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-0.1,'x=4.73')
legend('cos(x)','1/cosh(x)',3)
xn = linspace(0,4.73,50);
hold on
fill([xn,filplr(xn)],[1.cosh(xn),fliplr(cos(xn))],'c');
```

执行结果如图 6.11 所示。创建向量[1./cosh(xn) fliplr(cos(xn))],则可生成相连的多边形,由顶部曲线 $1/\cos(x)$ 和底部曲线[$\cos(x)$ 的逆向量]相互连接而成。新的 x 坐标向量[xn fliplr(xn)]对应新向量,由 x 的新值及其逆组合而成。

如果不使用填充命令,而用 20 条等间隔的垂直线填充,程序如下:

```
x = linspace(0,6,100);
```

```
plot(x,cos(x),'k-',x,1./cosh(x),'k--',[4.73,4.73],[-1,1],'k')
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-0.1,'x=4.73')
legend('cos(x)','1/cosh(x)','Location','South West')
xn=linspace(0,4.73,50);
hold on
xx=linspace(0,4.73,20);
plot([xx;xx],[cos(xx);1./cosh(xx)],'k-')
```

结果如图6.12(a)所示。

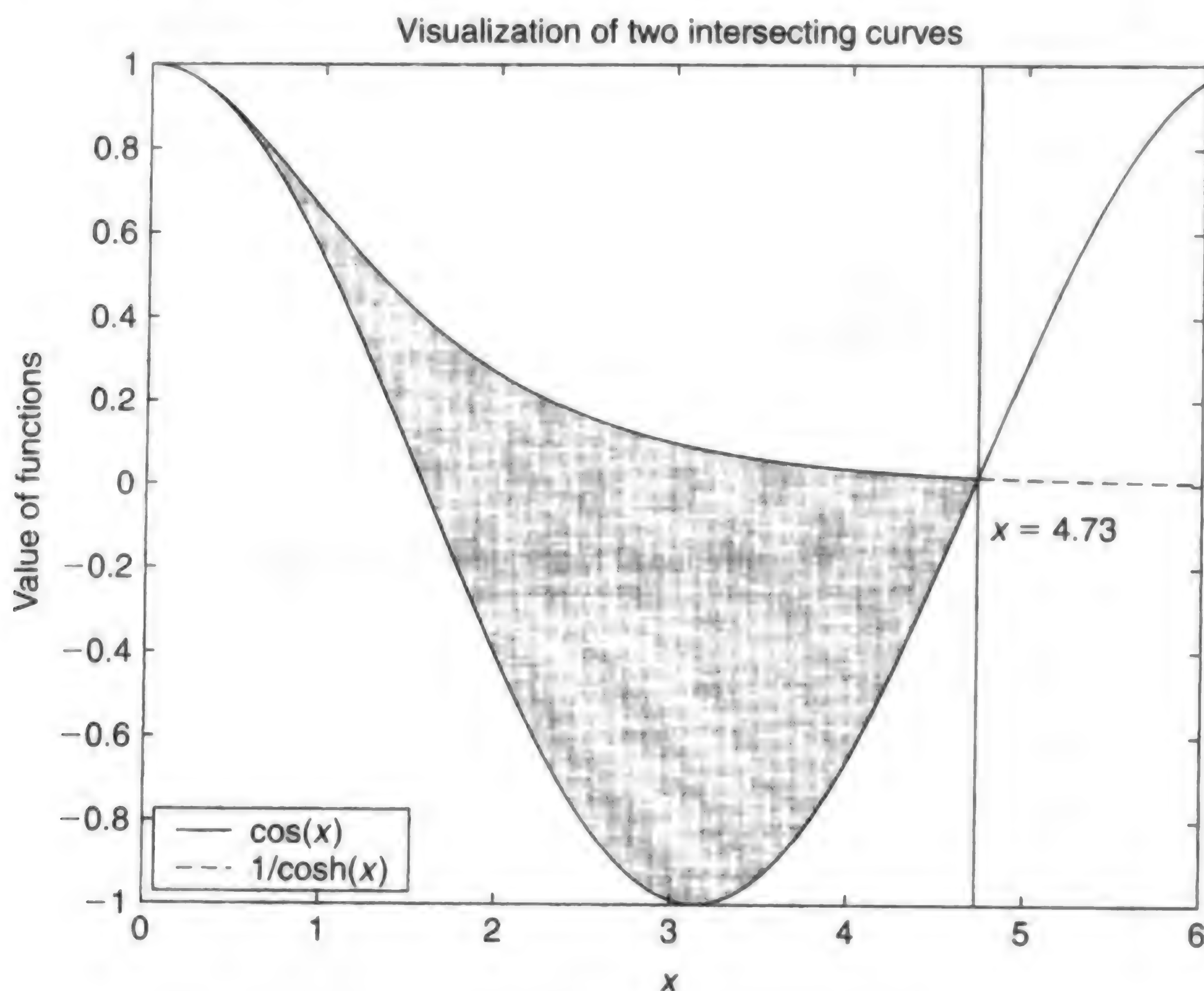


图6.11 填充图6.10中曲线间的区域

如果想画出20条等间隔的水平线,必须用反函数 arccos(x)和 arccosh(x)确定线段的终点。为此,可考虑两个区域。在第1个区域中 $y>0$,水平线右侧终点止于曲线 1/cosh(x);而在 $y<0$的区域中,水平线右侧终点则止于曲线 cos(x)。程序为:

```
x=linspace(0,6,100);
plot(x,cos(x),'k-',x,1./cosh(x),'k--',[4.73,4.73],[-1.1],'k')
hold on
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-0.1,'x=4.73')
legend('cos(x)',1/cosh(x)','Location','South West')
y1=linspace(1,0.01,10);
plot([acos(y1);acosh(1./y1)],[y1;y1],'k-')
y2=linspace(0.01,-1,10);
plot([acos(y2);pi+fliplr(acos(y1))],[y2;y2],'k-')
```

结果如图 6.12(b)所示。这两个程序结合起来可形成以下程序,并可生成如图 6.13 所示的阴影线效果。

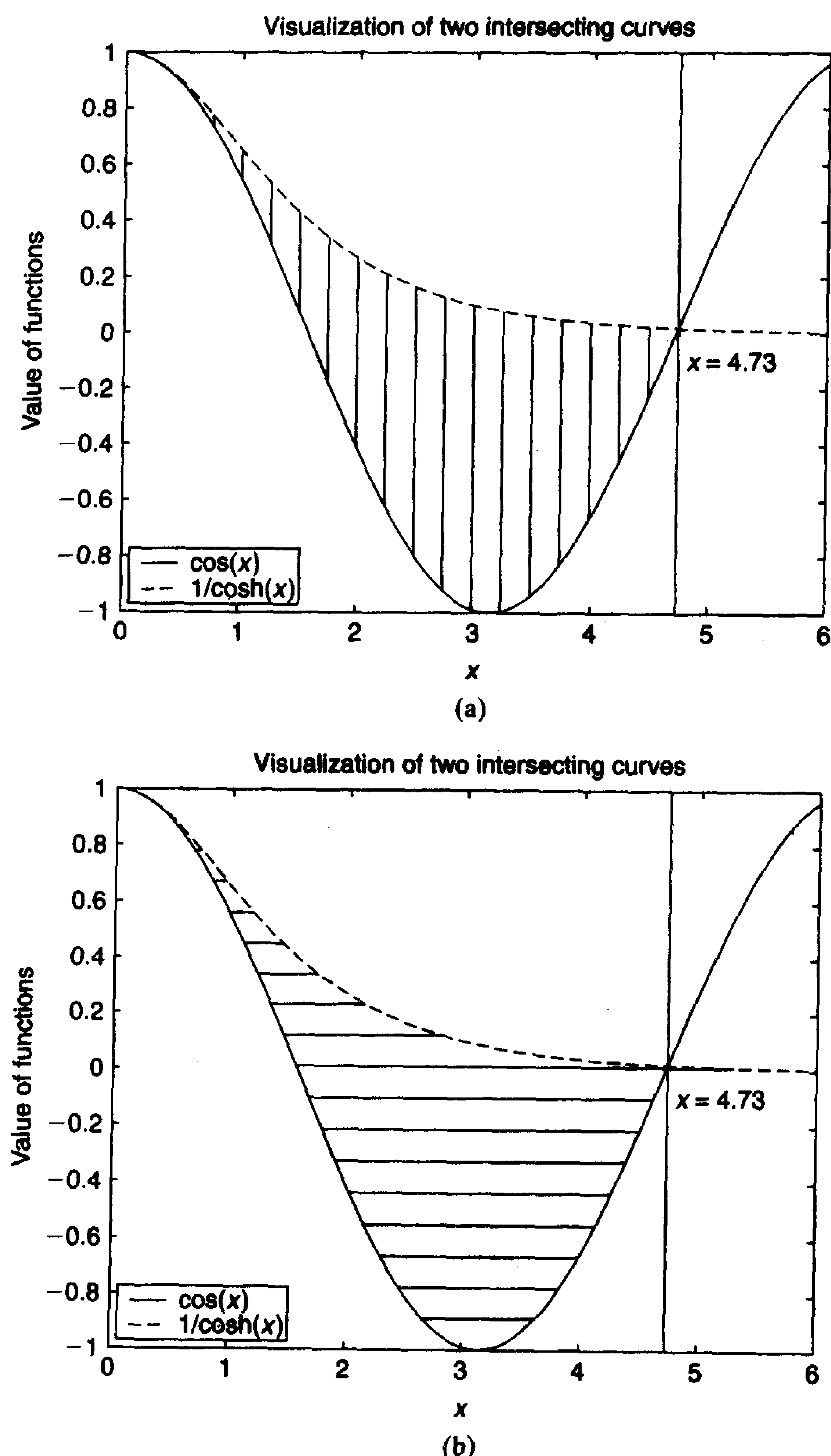


图 6.12 两条曲线相交区域用等间距线填充。(a) 垂直线;(b) 水平线

```
x = linspace(0,6,100);
plot(x,cos(x),'k-',x,1./cosh(x),'k--',[4.73,4.73],[-1,1],'k')
hold on
xlabel('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-0.1,'x = 4.73')
```

```
legend('cos(x)','1/cosh(x)','Location','South West')
xx = linspace(0,4.73,20);
plot([xx;xx],[cos(xx);1./cosh(xx)],'k-')
y1 = linspace(1,0.01,10);
plot([acos(y1);acosh(1.y1)],[y1;y1],'k-')
y2 = linspace(0.01,-1,10);
plot([acos(y2);pi+fliplr(acos(y1))],[y2;y2],'k-')
```

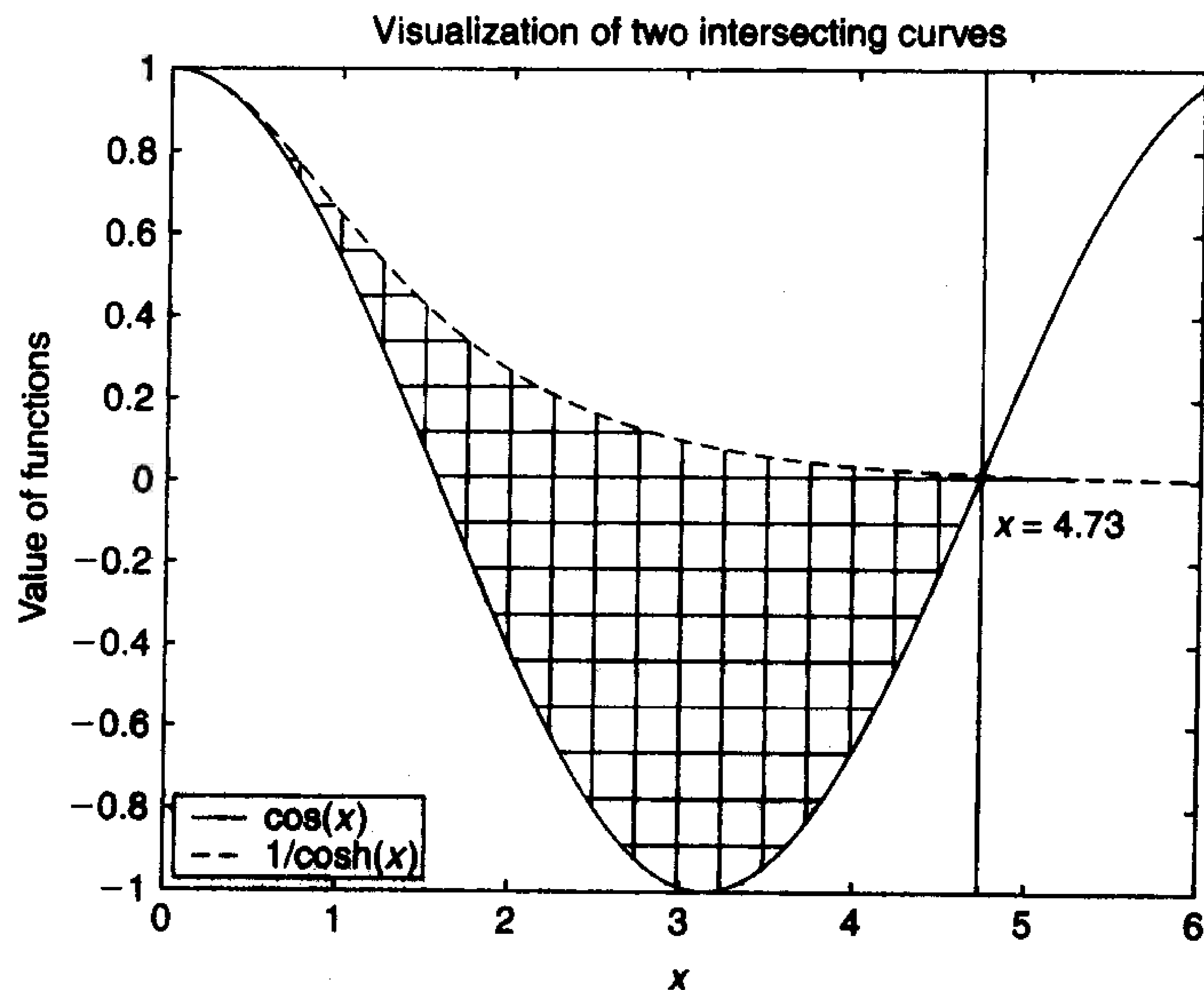


图6.13 在两条曲线相交区域填充后的阴影效果图

6.3.4 希腊字母、数学符号、下标和上标

MATLAB可以用大小写希腊字母、上标、下标和数学符号对图形进行标注。标注可通过xlabel,ylabel,text,legend和title完成。这些指令格式遵从*LaTeX*语言[①]。下列描述中,只有当符号处于撇号之间时方才有效,且所有字符都不能在命令窗口中(即使用disp函数时)使用。

下标由下划线(_)生成;上标由指数运算符(^)生成;希腊字母的生成如表6.5所示:先输入反斜杠(\),再输入字母的拼音。大写希腊字母仅须将拼音的首字符大写即可。然而,因为许多大写的希腊字母与大写的英文字母相同,所以表中仅列出其不同的部分,其他大写希腊字母可简单输入大写英文字母代替。

表6.5 大写和小写希腊字母以及部分数学符号

小写				大写		数学			
符号	句法	符号	句法	符号	句法	符号	句法	符号	句法
α	\alpha	ν	\nu	Γ	\Gamma	$\leqslant$	\leq	$\circ$	\circ
β	\beta	ξ	\xi	Δ	\Delta	$\geqslant$	\geq	$\ll$	\ll
γ	\gamma	o	o	Θ	\Theta	$\neq$	\neq	$\gg$	\gg

① 参见L. Lamport, LaTeX: A Document Preparation System, Addison-Wesley, Reading, MA, 1987年。

(续表)

小写				大写		数学			
符号	句法	符号	句法	符号	句法	符号	句法	符号	句法
δ	\delta	π	\pi	Λ	\Lambda	±	\pm	′	\prime
∈	\epsilon	ρ	\rho	Ξ	\Xi	×	\times	⇐	\Leftarrow
ζ	\zeta	σ	\sigma	Π	\Pi	∞	\infty	∠	\angle
η	\eta	τ	\tau	Σ	\Sigma	Σ	\sum	√	\surd
θ	\theta	υ	\upsilon	Υ	\Upsilon	∫	\int	#	\#
ι	\iota	ϕ	\phi	Φ	\Phi	÷	\div	$	\$
κ	\kappa	χ	\chi	Ψ	\Psi	~	\sim	%	\%
λ	\lambda	ψ	\psi	Ω	\Omega	←	\leftarrow	&	\&
μ	\mu	ω	\omega			↑	\uprrow	{	\{

除表 6.5 中所列出的字符外,可在字母数字字符前加入"\bf"使字符加粗。加入"\it"生成斜体字符。如果要恢复正常模式,则可在字母数字字符前加入"\rm"。

数学符号通过在其特殊拼音前加反斜杠(\)来生成。一些常用符号也在表 6.5 中给出。常用语法是在一对单引号间放置一组相关指令。当一组符号要放置在一起时,例如在指数中显示的表达式,则可将其放在一对大括号({})之内。下面举例说明这些过程。

当 $\beta=3, 1\leqslant\Omega_1\leqslant 2$ 时,计算以下函数并对相应图形进行标注:

$$g_2=\cos(4\pi x)e^{-(1+x^{\beta})}$$

程序为:

```
x=linspace(1,2);beta=3;
plot(x,cos(4*pi*x).*exp(-(1+x.^beta)),'k')
title('\itg_{\rm2}\rmversus \itx \rmfor \it \beta \rm=3')
ylabel('\itg_{\rm2}')
xlabel('\itx')
text(1.2,0.08,'itg_{\rm2}\rm=cos(\itx \rm) \ite^{\rm-(1+ \itx^\beta \rm)}')
```

结果如图 6.14 所示。

6.3.5 轴、曲线和文本属性的改变

MATLAB 能够改变图形元素的所有属性。对于曲线,将讨论其线宽和曲线颜色;对于轴标注、图形标题、添加文本和图注将讨论其字体、排列、大小、类型、字符和颜色;对于轴,将讨论其线宽和文本属性。

轴和绘制曲线的默认线宽为 0.5 pt。轴标注、添加文本和图形标题默认字体大小为 10 pt。轴标注、数字、图形标题和图注的默认字体名为 Helvetica。

要改变曲线和文本的属性,则对于函数 xlabel,ylabel,title 和 text 可添加关键字及其数值作为这些函数的参数:

```
xlabel(s,'Key word', Key Word Value,...)
ylabel(s,'Key word', Key Word Value,...)
```

```
title(s,'Key word', Key Word Value,...)
text(x,y,s,'Key word', Key Word Value,...)
```

其中,'Key word'为包含关键字的字符串,*KeyWordValue* 是与关键字相对应的字符串或数值。表 6.6 给出了文本所处位置的关键字及其值。表 6.7 给出了有关文本属性的其他关键字及其值。

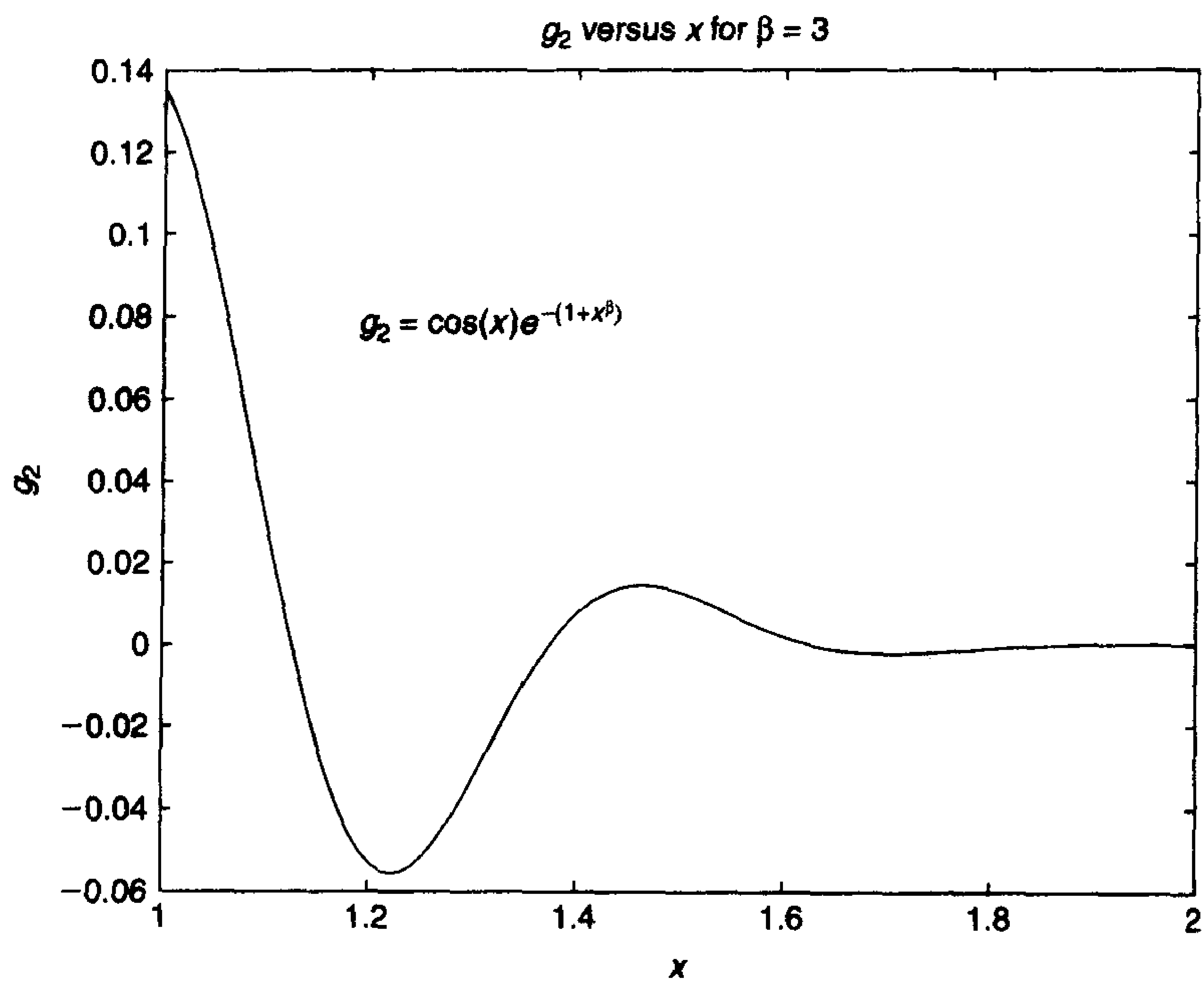


图 6.14 用上标、下标和希腊字母对图形进行标注

表 6.6 文本位置关键字及其属性

关键字	关键字的值	示例
'Horizontal Alignment'	'Left' 'Center' 'Right'	Left Center Right
'Vertical Alignment'	'Top' 'Middle' 'Bottom'	Top Middle Bottom
'Rotation'	0 ~ 360°或 −180° ~ +180°	90 180 or −180 · 0 or 360 −90 or 270

表 6.7 其他文本关键字及其属性

关键字	关键字值
'Linewidth'	number > 0 (default = 0.5)
'FontSize'	number > 0 (default = 10)
'FontName'	'Courier'
	'Helvetica' (default)
	'Times' (Times Roman)
'Color'	'Letter from 2nd column of Table 6.1'
'FontWeight'	'Normal' (default)
	'Bold'

要设置 plot 函数绘制的曲线和 legend 函数显示的文本属性,需要获得函数的句柄。利用 set 函数和 gca 函数可改变轴线线宽和轴标注。gca 函数可获得当前轴的句柄。set 函数设置与句柄相关的特定属性。为获得 legend 和 plot 函数的句柄,可输入:

```
hdl = plot(...);
[lh,lo] = legend(...);
```

其中,plot 函数和 legend 函数的运行方式已经讨论过。*hdl* 为绘制曲线的句柄。*lo*(1)为 legend 函数显示文本的句柄。注意,在上述语句末尾添加了分号,以避免句柄数值显示在命令窗口。set 函数为:

```
set(hdl,'Key word',KeyWordValue, ...)
```

其中,*hdl* 为句柄,'Key word'为包含关键字的字符串,*KeyWordValue* 为与关键字对应的字符串或数值。表 6.8 中给出了两个可改变线属性的关键字。

表 6.8 线的关键字和属性

关键字	属性
'Linewidth'	number > 0 (default = 0.5)
'Color'	'Letter from 2nd column of Table 6.1'

修改生成图 6.10 的程序,使得图形具备以下特性:

图形标题:字号 14 pt、Courier 字体、加粗
x 轴标注:字号 14 pt、Times Roman 字体、斜体加粗
y 轴标注:字号 14 pt、Helvetica 字体
添加文本:字号 12 pt、标准算术符
图注:字号 14 pt、Helvetica 字体、红色
轴线:线宽 1.5 pt
轴标注:字号 14 pt、Helvetica 字体
cos(x)曲线:线宽 4 pt
cosh(x)曲线:线宽 2.5 pt
x = 4.73 处的垂线:线宽 0.25、蓝色

修改后的程序为：

```
x = linspace (0,6,100);
hc = plot(x,cos(x),'k-');
hold on
hch = plot(x,1./cosh(x),'k--');
hsl = plot([4.73,4.73],[-1,1],'k');
xlabel(' \it \bfx','FontSize',14,'FontName','Times')
ylabel('Value of functions','FontSize',14)
title (' \bfVisualization of two intersecting curves','FontName','Courier',...'FontSize',14)
text(4.8,-0.1,' \itx \rm = 4.73','FontName','Times','FontSize',12)
[a,b] = legend ('cos(x)','1/cosh(x)','Location','SouthWest');
set(hc,'LineWidth',4)
set(hch,'LineWidth',2.5)
set(hsl,'LineWidth',0.25,'Color','b')
set(b(1),'FontSize',14,'Color','r')
set(gca,'FontSize',14,'LineWidth',1.5)
```

结果如图6.15所示。

对文本、曲线和轴属性的修改也可以通过直接在图形窗口顶部选择适当的图标，或在"*Tools*"菜单中选择相应的操作项来完成。图形一旦改变，可以保存为Windows的图元文件或者作为一段程序保存在".M"文件中。

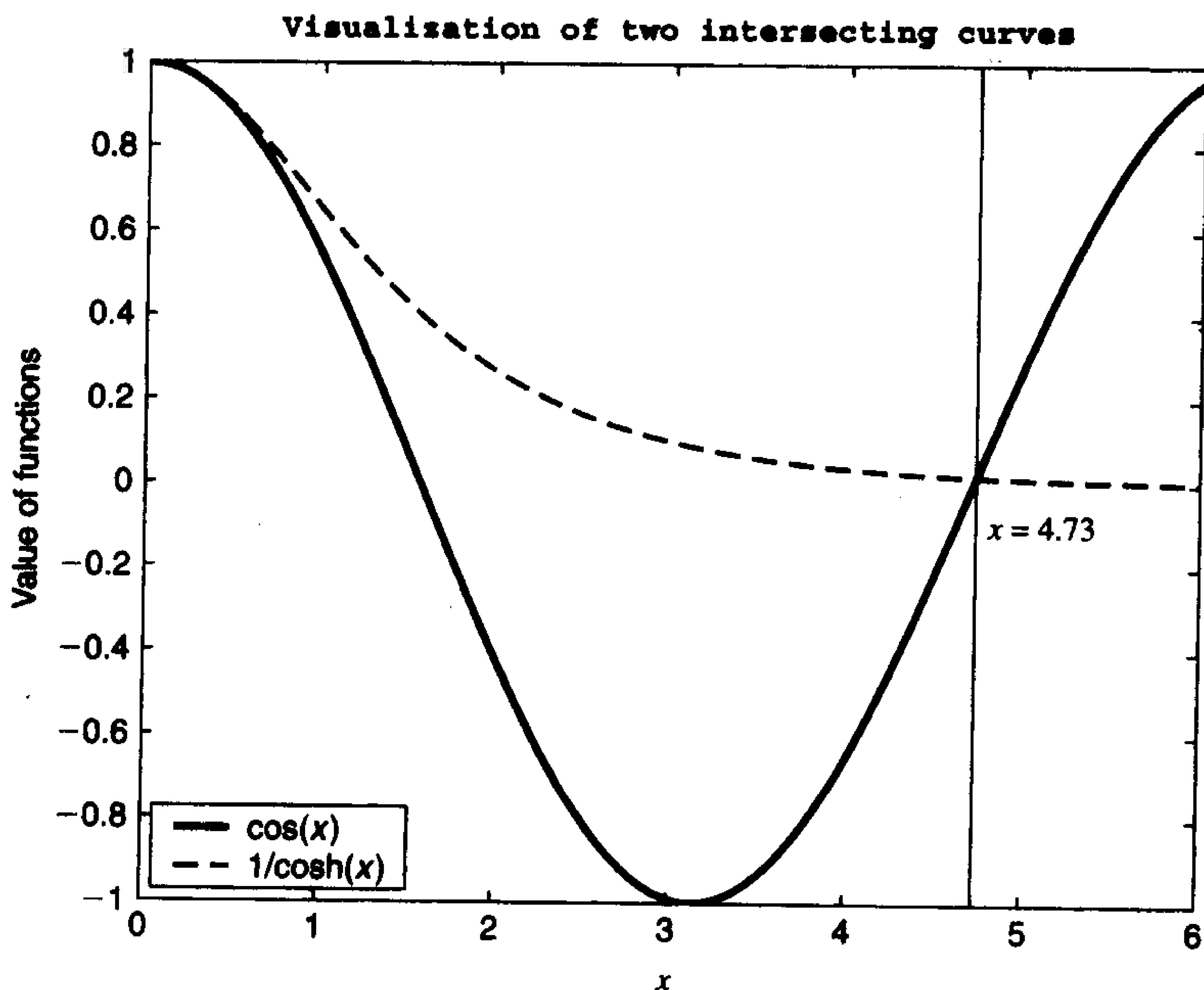


图6.15 字体大小、字型和线宽的更改

下面举例说明上述结果。

例6.1 重复曲线：在 $0 \leqslant x \leqslant m\pi$ 范围内绘制 $\cot(x)$ 曲线

下面将在 $0 < x < m\pi$ 范围内显示 $\cot(x)$，其中 m 由用户选择，可以是1,2,3,…，6。$m=4$ 时的图形如图6.16所示。y 轴的上下限为 ± 8，x 轴的取值范围从0到 $m\pi$。因为 $\cot(0) = \infty$，$\cot(\pi) = -\infty$，所以 x 的取值不包括0和 π。因为余切函数的周期为 $m\pi$，所以只须

画出 $0<x<\pi$ 间的函数图形,然后将其右移 $x+(m-1)\pi$ 即可得到 $(m-1)\pi<x<m\pi$, $m>1$ 区域内的函数图形。程序如下:

```
m = input('Enter an integer from 2 to 6):');
the = linspace(0.12,pi - 0.12,50);
ct = cot(the);
hold on
for n = 1:m
  plot([2 * n - 1] * pi/2,(2 * n - 1) * pi/2],[ - 8/40,8/40],'k - ')
  if n == 1
    text((2 * n - 1) * pi/2 - pi(8 * m),0.6,' \ pi/2')
  else
    text((2 * n - 1) * pi/2 - pi(8 * m),0.6,[num2str((2 * n - 1),2)' \pi/2'])
  end
  if n == m
    plot(the + (n - 1) * pi,ct,'k - ')
  else
    plot(the + (n - 1) * pi,ct,'k - ',[n * pi,n * pi],[ - 8,8],'k- -')
  end
end
plot([0,m * pi],[0,0],'k - ')
axis([0,m * pi, - 8,8])
xlabel('\itx \rm(radians)')
ylabel('cot(\itx \rm)')
title(['Cotangent function from 0 to 'num2str(m,1),'\pi'])
box on
```

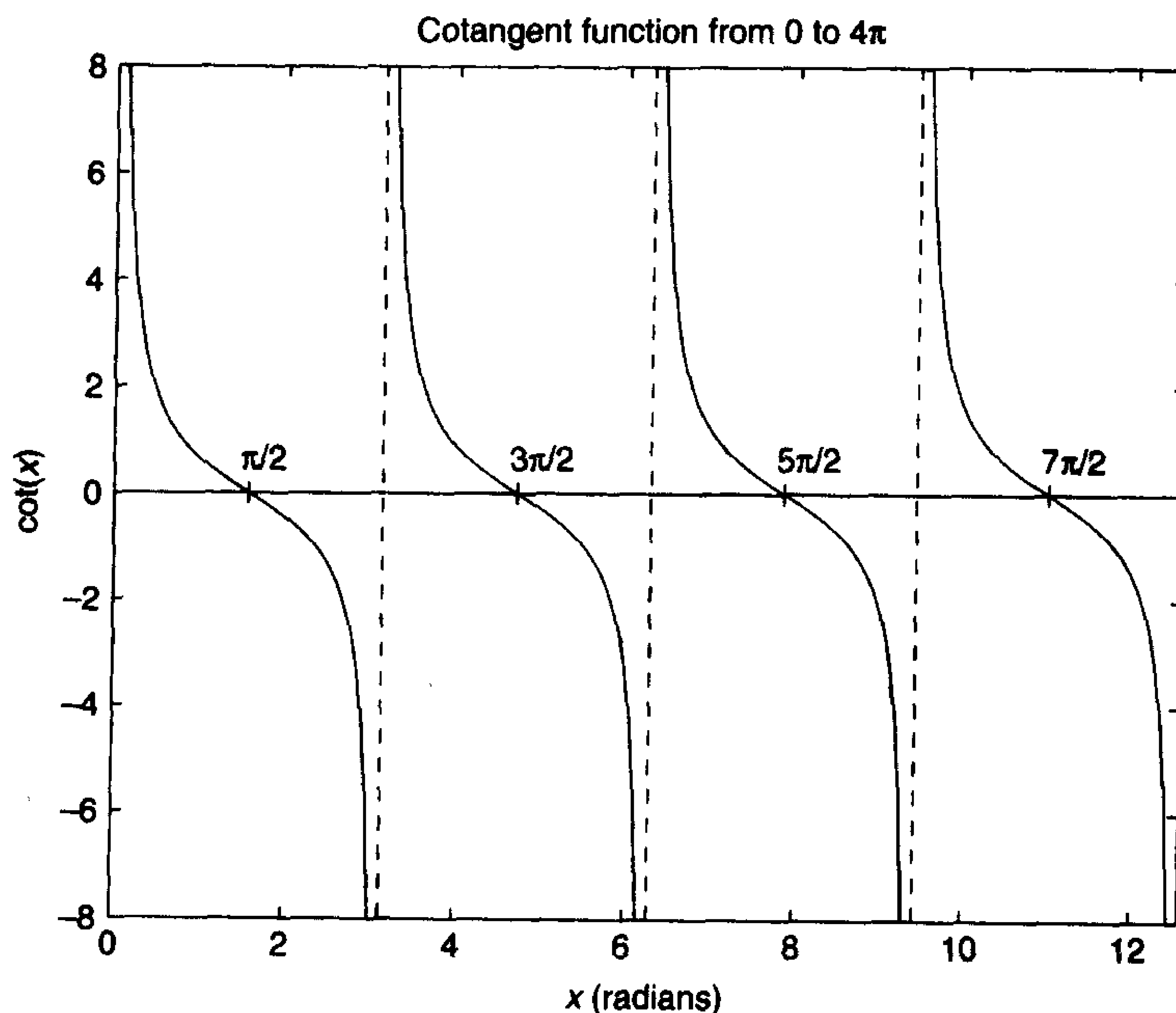


图 6.16　余切函数

可以看出几乎所有的程序行都用于函数图形的标注,仅有两行语句用于绘制 cot(x)的函数图形。

例 6.2 极坐标图形:声源的远域辐射模式

无限大挡板内的圆形活塞以频率 f 振动,距离其中心极远处的标准声压由下式给出:

$$p(r,\theta)=\left|\frac{J_1(ka\theta)}{ka\theta}\right| \quad ka^2<<r \quad 和 \quad a<<r$$

其中,r 为距离活塞圆心的极半径,θ 为 r 与挡板平面之间的角度,k 为波数,a 为活塞半径,$J_1(x)$为一类 1 阶贝塞尔函数。波数是频率为 f 的声波波长的倒数,因此,ka 是无量纲的。该模型是对扩音器声波角扩散的一个很好的近似。

创建一个标准辐射模型的极坐标,其中,$ka=6\pi$, $-\pi/2<\theta<\pi/2$。该例同时也可说明 polar 函数的用法,可以实现在极坐标中绘制结果曲线。程序如下:

```
theta = linspace( - pi/2,pi/2,300);
p = abs(besselj(1,6 * pi * theta)./(6 * pi * theta);
polar(theta, p/max(p))
```

以上程序结果如图 6.17(a)所示。注意,θ 的值使 $ka\theta\neq0$。max 函数得到向量 p 的最大值,因此 p/max(p)的比值是标准的辐射模型,其最大值为 1。放大区域为 $r<0.1$ 附近,放大操作有两种处理方法:其一是单击图形窗口的"放大"图标放大该区域,另一种方法是修改程序并设定笛卡儿坐标系的观测区域。在命令窗口中输入:

```
v = axis
```

结果为:

```
v =
   -1.0000    1.0000    -1.1500    1.1500
```

即 x 轴取值范围为 ±1 范围内,y 轴取值范围为 ±1.15 范围内。由此,可用以下语句确定视图区域:

```
axis([ -.02,0.15, - 0.05,0.05])
```

修改后的程序为:

```
theta = linspace( - pi/2, pi/2, 300);
p = abs(besselj(1,6 * pi * theta)./(6 * pi * theta));
polar(theta, p/max(p))
axis([ -.02,0.15, - 0.05,0.05])
```

执行结果如图 6.17(b)所示。

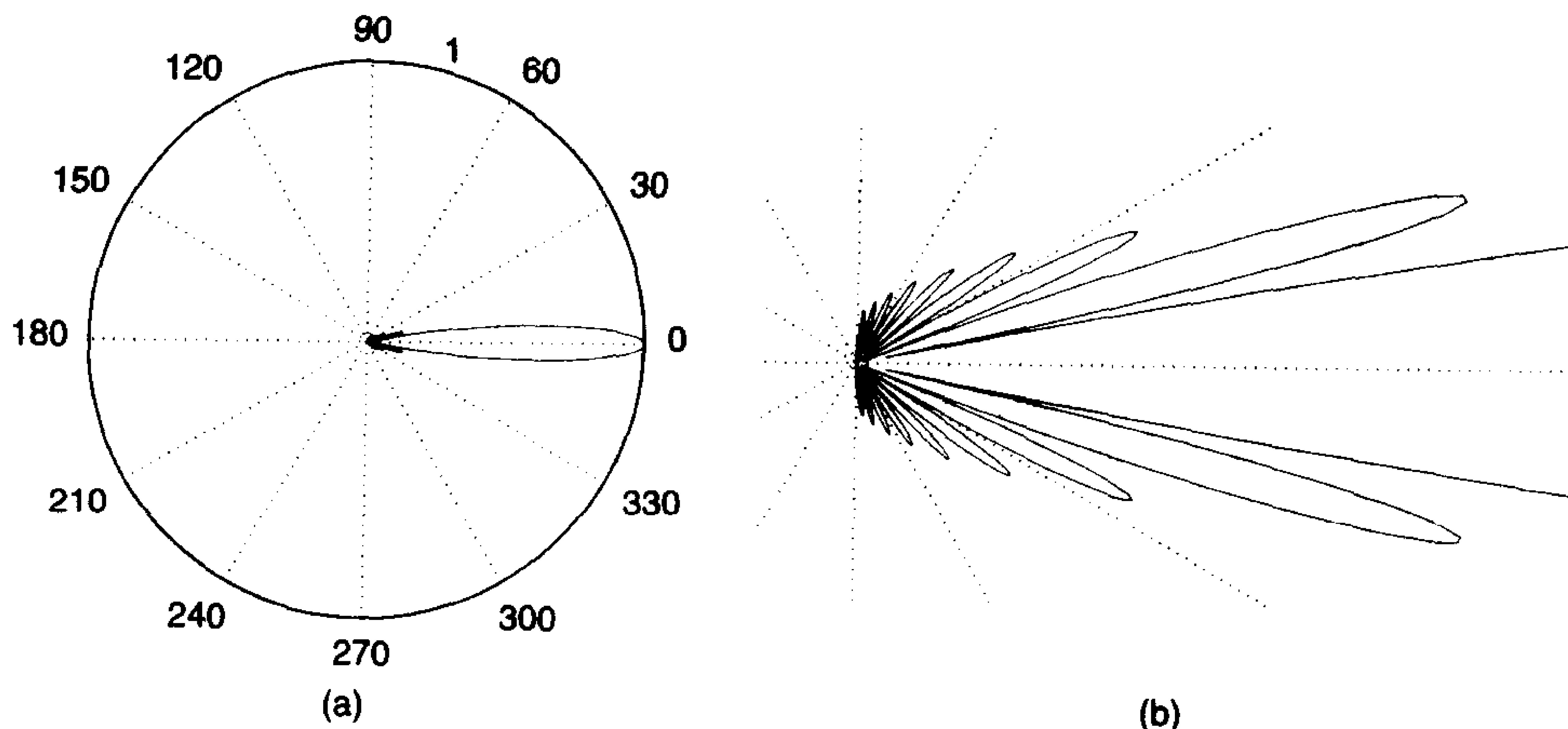


图 6.17 (a) 幅射模式的极坐标表示;(b) 放大区域

例 6.3 多个图形的绘制:周期脉冲序列和单脉冲的频谱图

如图 6.18(a)所示,一个周期矩形脉冲序列,脉宽为 d,周期为 T,该信号可表示为(参见例 2.11):

$$g(t) = \frac{f(t)T}{Ad} = c_0 + 2\sum_{n=1}^{\infty} c_n \cos(n\omega_0 t)$$

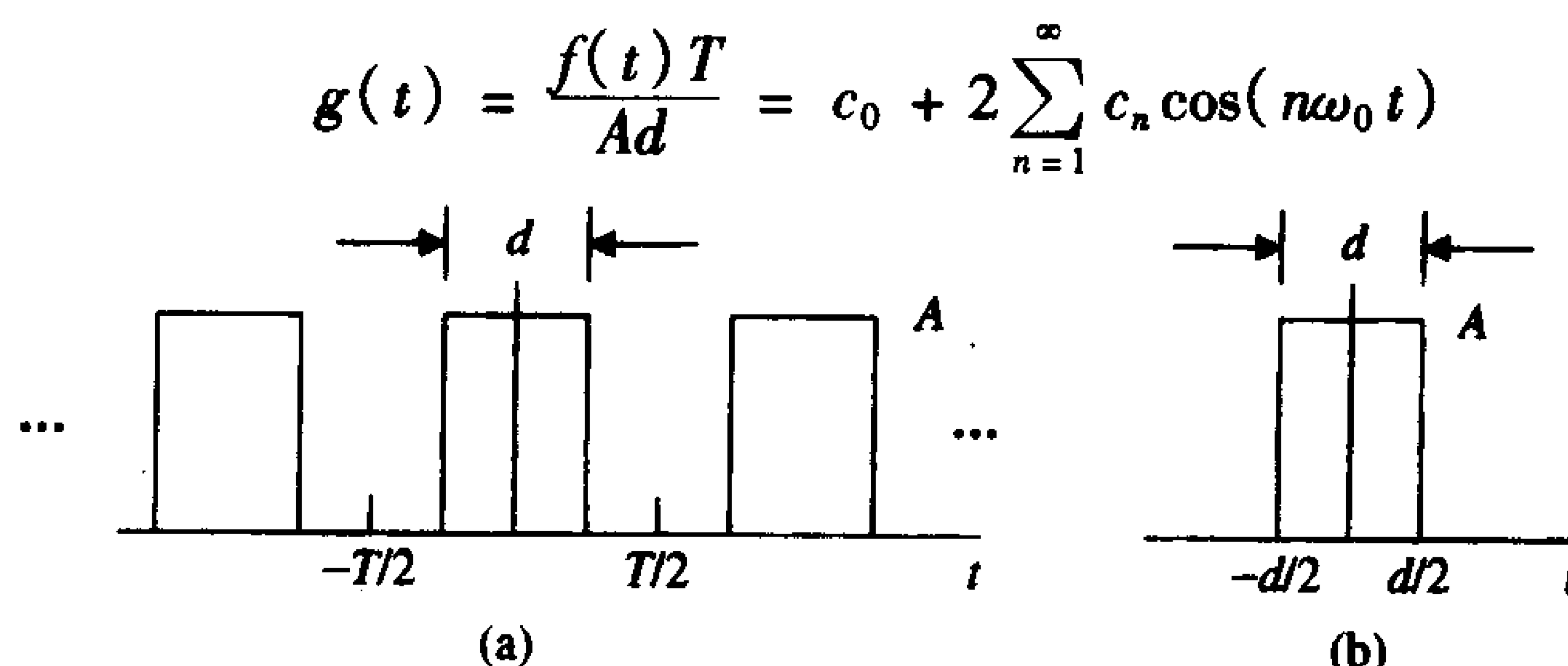


图 6.18 (a)周期矩形脉冲序列;(b) 单矩形脉冲

其中,$\omega_0 = 2\pi/T$,且,

$$c_0 = 1$$

$$c_n = \frac{\sin(n\pi d/T)}{(n\pi d/T)} \quad n = 1,2,\cdots$$

c_n 为信号谐波的幅值,其模 $|c_n|$ 为 n 的函数,在图中画出该函数,则称为信号的幅度谱。频谱图中只有 ω_0 处(即 $n\omega_0$ 处)谐波分量不为 0,其余频率的谐波分量均为 0。由式中可以看出,当 $c_n = 0$ 时,

$$\frac{n\pi d}{T} = m\pi \quad 或 \quad n = \frac{m}{(d/T)}$$

另一方面,如果有一单脉冲,如图 6.18(b)所示,其频谱为:

$$F(0) = 1$$

$$F(\omega) = \frac{G(\omega)}{Ad} = \frac{\sin(\omega d/2)}{\omega d/2} \quad \omega > 0$$

$G(\omega)$的绝对值称为振幅密度频谱。

先创建两幅图形,一幅是 $|c_n|$ 对 n 的变化图,其中 $n = 1,2,\cdots,30$ 且 $d/T = 0.1$;第二幅是 $|F(\omega d)|$ 对 ωd 的变化图,其中 $0 \leqslant \omega d \leqslant 6\pi$。为创建第一幅图,使用 6.2.2 节中的方法画一系列未连接的直线。画出的两幅图形,一幅在另一幅的上方。注意,不要使除数为 0。以下程序的结果如图 6.19 所示。

```
n=1:30;dT=0.1;
cn=[1 abs(sin(dT*pi*n)./(dT*pi*n))];
n=[0 n];
subplot(2,1,1)
polt([n;n],[zeros(1,length(cn);cn)],'k')
xlabel('Harmonic number ( \itn \rm)')
ylabel('|itc _ n \ rm|')
text(15,.9,['d/T='num2str(dT)])
title('Amplitude spectrum')
w=pi/5:pi/5:6*pi;
subplot(2,1,2)
plot([0 w],[1 abs (sin(w/2)./(w/2))],'k')
axis([0,6*pi,0,1])
```

```
xlabel ('\it \omegad')
ylabel('|F(\it \omegad)|')
text(3*pi,.9,'Single pulse')
title('Amplitude density spectrum')
```

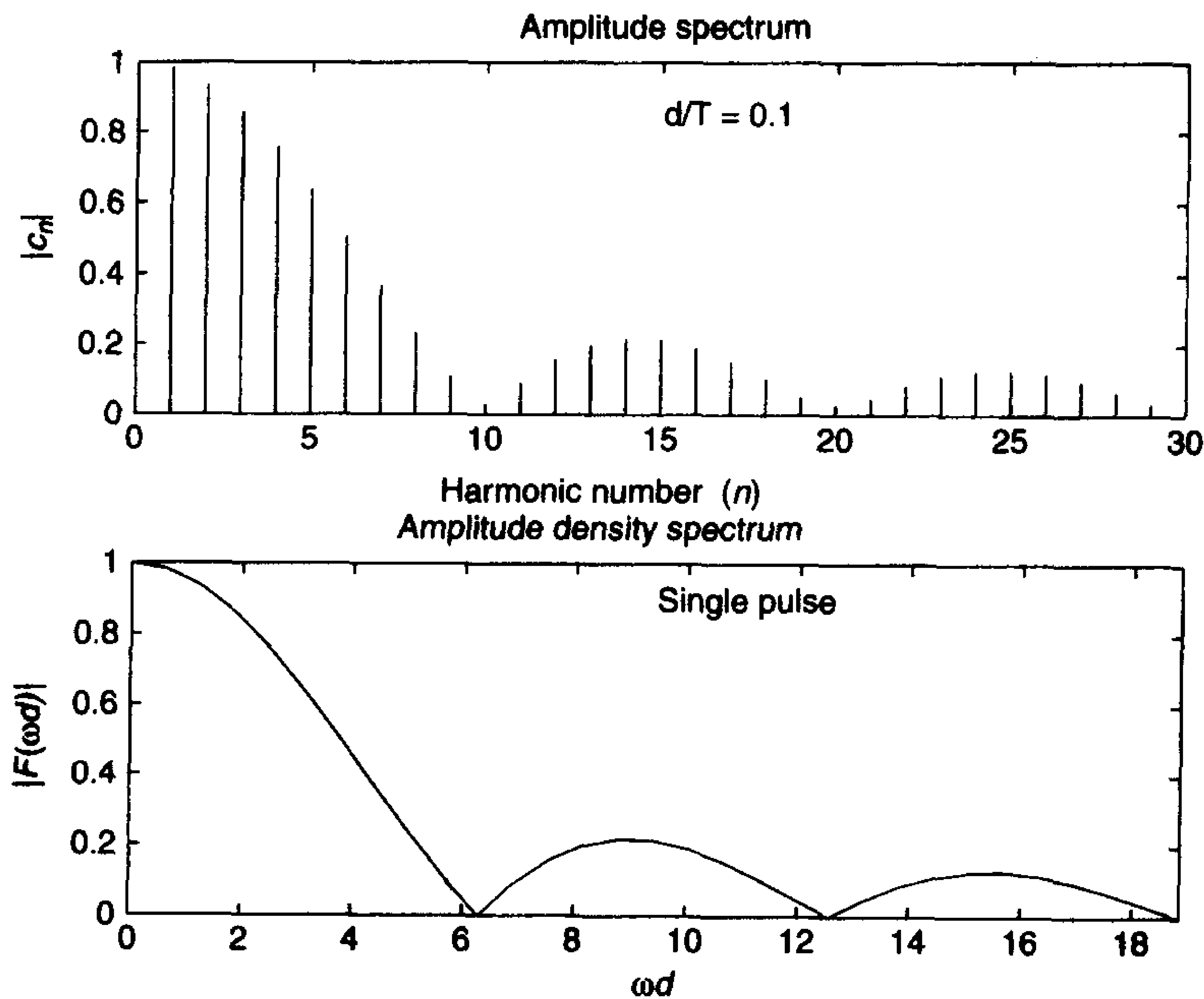


图 6.19 (a) 图 6.18(a)所示的周期脉冲频谱;(b) 图 6.18(b)所示的单脉冲频谱

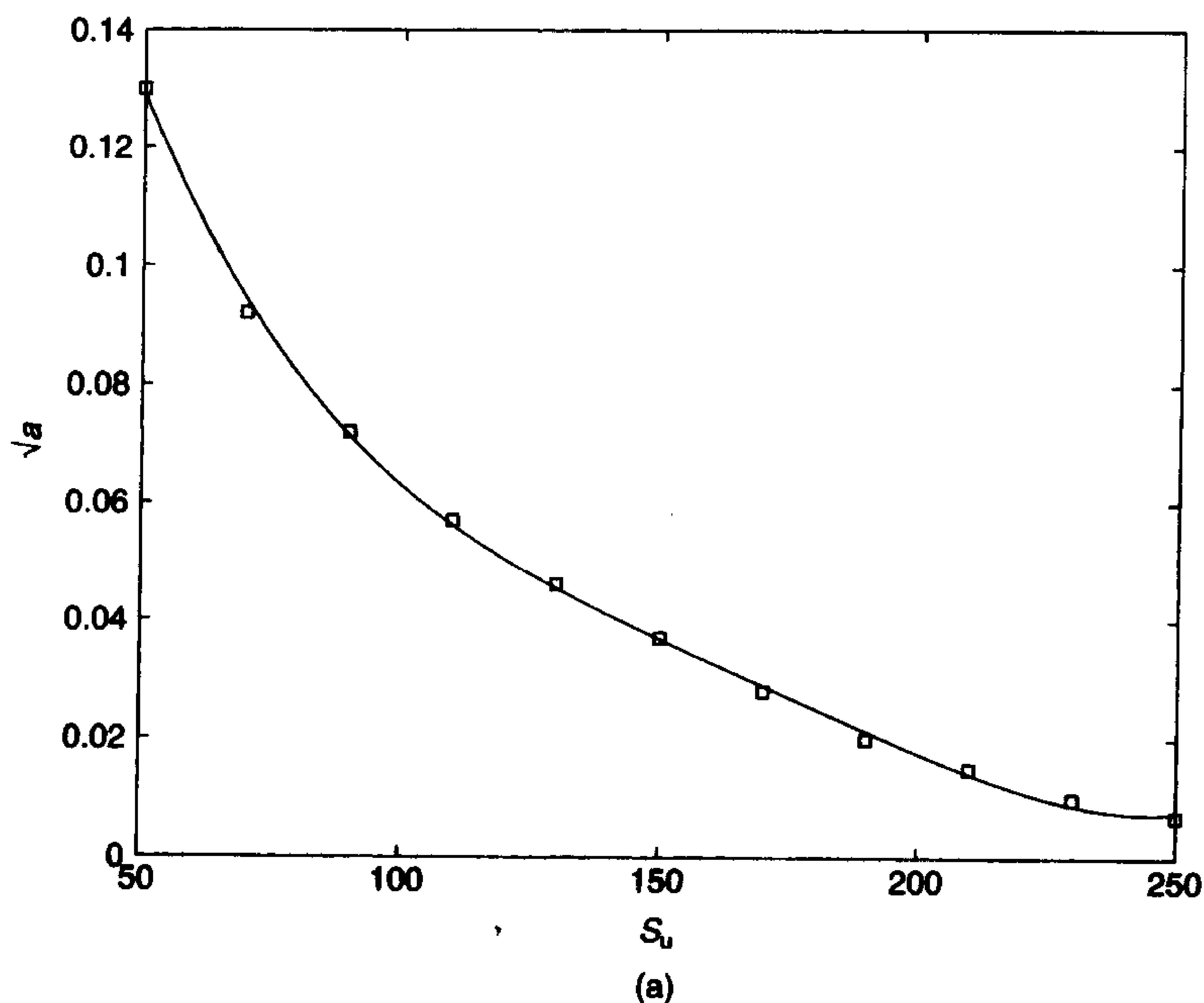
可采用以下方法避免除数为 0 的操作:谐波次数向量最初定义为 $n=1:30$,这样用点除即可方便地计算出在此范围内的$|c_n|$值,然后分别创建包括 $n=0$ 和 $c_0=1$ 的两个向量,如第二和第三条曲线所示。采用同样的方法可以计算出$|F(\omega)|$的值。该方法不需要任何编程逻辑。

例 6.4 多条曲线的绘制:钢的槽口敏感性

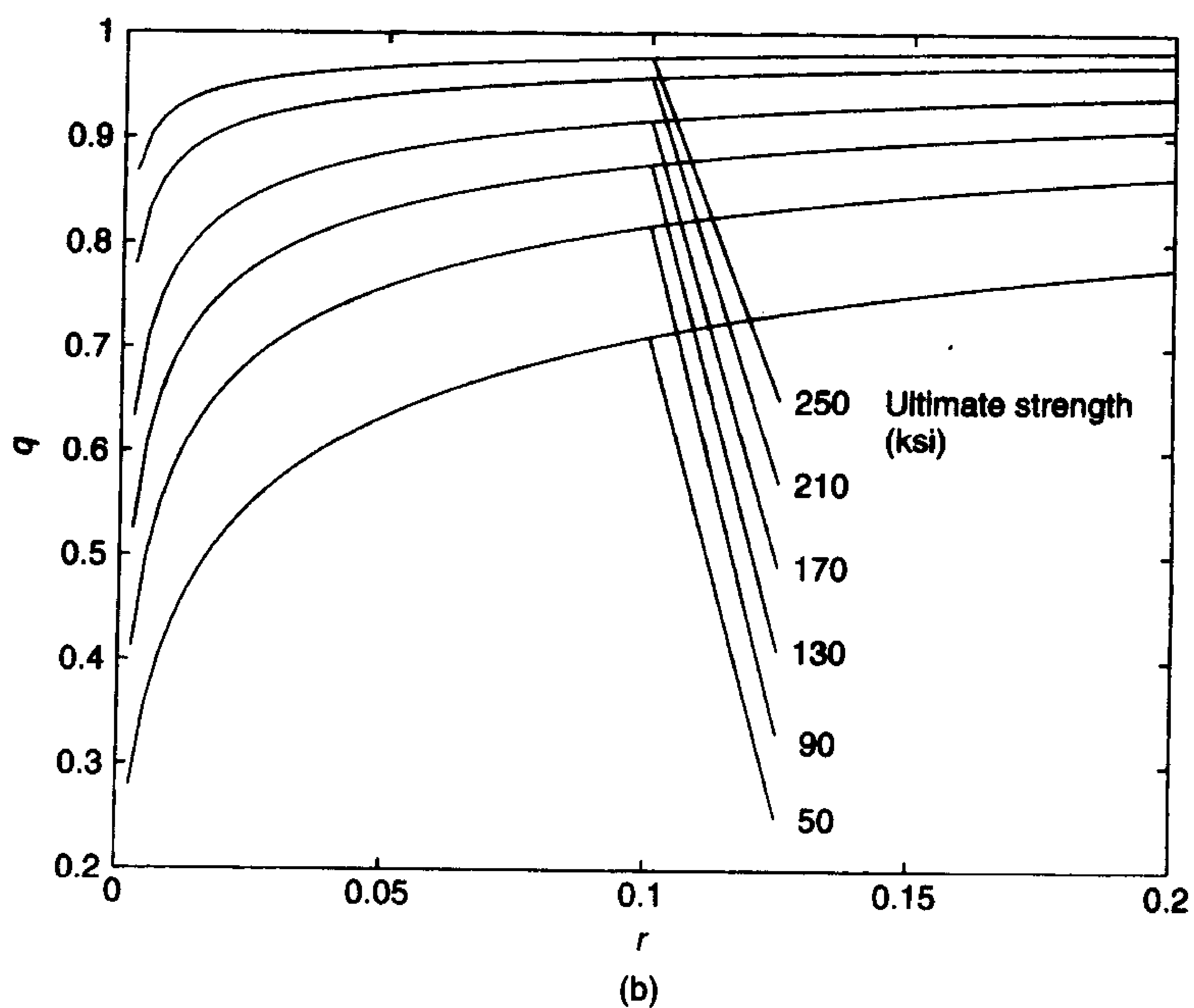
参考 5.1 节的例子,在 $50\leq S_u\leq 250$ 和 $0<r<0.2$ 的范围内画出槽口敏感性常数 q 的曲线。为提高程序的可读性,对要填充的数据创建一个函数。程序由两部分组成。第一部分求得四次多项式的系数,并画出数据点及这些点处的多项式值。第二部分用多项式创建曲线族,这些曲线为钢的极限强度 S_u 取不同值时槽口敏感性常数 q 对槽口半径的关系曲线。程序执行结果如图 6.20 所示。

```
Su = linspace (50,250,50); skip = 1:2:11;   loc = 0.25:0.08:0.65;
ncs = NeuberData;
p = polyfit(ncs(:,1),ncs(:,2),4);
figure(1)
plot(Su,polyval(p,Su),'k',ncs(:,1),ncs(:,2),'ks')
xlabel('\itS_u')
ylabel('\surd \ita')
figure(2)
[s,r] = meshgrid (ncs(skip,1),linspace(0.0025,0.2,80));
notch = inline('1./(1+polyval(p,s).//sqrt(r))','p','s','r');
```

```
plot(r,notch(p,s,r),'k')
hold on
plot([repmat(0.125,1,6);repmat(0.1,1,6)],[loc;notch(p,ncs(skip,1)',0.1)],'k')
text(repmat(0.13,1,6),loc,num2str(ncs(skip,1)))
text(0.145,0.65,'Ultimate strength')
text(0.145,0.62,'(ksi)')
xlabel('\itr')
ylabel('\itq')
```



(a)



(b)

图 6.20 (a)利用四阶拟合的钢的纽博尔(Neuber)常数;
(b)钢的槽口敏感性对槽口半径r的函数曲线

其中，**NeuberData** 在例 5.1 中给出。

meshgrid 函数生成了两个(80×6)数组，其中行是 r 的值，列是 S_u 的值。由于 inline 函数中的表达式 **notch** 使用点标记，所以可以在数组中得到合适的参数。最后一个 plot 函数放置曲线标识线，如前所述，画出一系列未连接的直线。但必须将列向量 $ncs(:,1)$ 转换为行向量。为了在线段末端加上文字注释，可创建 x, y 的坐标向量，其相应文字由 num2str 函数将 ncs(:,1)的第 1 列转换为字符串而得到。将其第 1 个下标加 2 可选择 ncs 的另一元素。

例 6.5 旅行到 N 点的最小距离总和

在笛卡儿坐标系上半部分考虑任意 N 点，且已知原点到每点的距离。如果设定每点的坐标为(x_j, y_j)，其中，$j=1,2,\cdots,N$，则不包括原点在内的任意两点之间的距离为：

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i+y_j)^2} \quad i\neq j \quad i,j=1,2,\cdots,N$$

从原点出发，历经各个点，最后返回到原点的距离总和 d_T 为：

$$d_T = d_{o1} + \sum_{j=2}^{N} d_{j-1,j} + d_{No} \tag{6.1}$$

其中，d_{o1} 为原点到第 1 点的距离，d_{No} 为第 N 点到原点的距离。原点坐标为(0,0)。

为求得最小的距离总和，必须确定经过 N 个不同点的次序，即要对式(6.1)中 j 的次序进行 $N!$ 次重排以重新组合 N 和 $d_{j-1,j}(j=2,\cdots,N)$。为此，在式(6.1)中引入$(N!\times N)$阶矩阵 $p_j(k)$，其中，$j=1,2,\cdots,N, k=1,2,\cdots,N!$。矩阵中每一行为重新排列后的 N 的位置。例如，$N=2$ 时则经过各点有两种可能的途径：其一是从原点到位置 1，然后到位置 2，再返回到原点。另一种途径是从原点到位置 2，然后到位置 1，再返回到原点。即有 $p_1(1)=1, p_2(1)=2, p_1(2)=2, p_2(2)=1$。则式(6.1)变为：

$$d_T(k) = d_{o,p_1(k)} + \sum_{j=2}^{N} d_{p_j(k)-1,p_j(k)} + d_{p_N(k),o} \quad k = 1,\cdots,N!$$

$d_T(k)$最小时，k 值确定 $p_j(k)$的阶次。

数据排列由下列语句完成：

```
perms(v)
```

其中，v 为行向量。如 $v=[3,15,7]$，则以下语句

```
p=perms([3,15,7])
```

的执行结果为：

```
p=
    7   15    3
    7    3   15
   15    7    3
   15    3    7
    3   15    7
    3    7   15
```

求经过 N 点的距离总和 $d_T(k)$ 最小时的各点的访问次序，并绘制各点及访问次序图形。假设 $N=6$，各点的坐标由数组 xx 和 yy 给出，且第 1 对坐标表示位置 1，依次类推。函数

dT 计算距离总和。程序为:

```
function MinimumDistance
xx = [ - 6, - 6, - 1,0,2,2];
yy = [7,2,5,9,7,2];
N = length(xx);
p = perms(1:N);
for k = 1:factorial(N)
  TotalDist(k) = dT(xx,yy,p,k);
end
[SmalDist,indx] = min(TotalDist);
BestOrder = p(indx,:);
plot([xx],[yy],'sk')
v = [ - 10,10,0,10];
hold on
xplot = [zeros(1,5);[v(2),v(4)/tan(pi/3),0, - v(4)/tan(pi/3),v(1)]];
yplot = [zeros(1,5);[v(2) * tan(pi/6),v(4),v(4),v(4), - v(1) * tan(pi/6)]];
plot(xplot ,yplot,'k--')
grid on
for n = 1:N
  text(xx(n) + v(2)/25,yy(n),num2str(n))
end
plot([0,xx(BestOrder)],[0,yy(BestOrder)],'k - ','LineWidth',2)
title(['Minimum distance = 'num2str(SmalDist,4)])
xlabel ('x')
ylabel('y')
function distance = dT(x,y,p,k)
dist = inline('sqrt(diff(x).^2 + diff(y),^2)','x','y');
N = length(x);
dista = [dist([0,x,(p(k,1))],[0,y(p(k,1))]),...
       dist(x(p((k,1:N)),y(p(k,1:N))),...
       dist([x(p(k,N)),0],[y(p(k,N)),0])];
distance = sum(dista);
```

执行结果如图 6.21 所示。

6.4 应用函数一览

表 6.1 给出了线和点的属性定义。表 6.5 给出了用于显示希腊字母和数学符号的关键字。表 6.6 给出了有关文本位置的关键字。表 6.7 给出了其他文本属性关键字。表 6.8 给出了线属性关键字。表 6.9 中,将介绍本章应用到的绘图函数。

表 6.9 第 6 章应用的 MATLAB 函数

MATLAB 函数	说明
axis	改变轴的外观和比例
axis equal	设置轴使得各个方向的高宽比单位相同
axis on/off	打开或关闭轴的显示
bar	创建柱形图

(续表)

MATLAB 函数	说明
box on/off	打开或关闭轴边界显示
factorial	计算 $N!$，其中 N 为正整数
figure	创建视图
fill	填充多边形
gca	获得当前图形当前轴的句柄
grid on/off	打开或关闭二维或三维图形中的栅格
hold on/off	打开时在图形中添加新的图形对象，关闭时替换图形对象
legend	在图形中显示图注
loglog	创建以对数为刻度标度的 x 轴和 y 轴
perms	计算所有可能的排列值
plot	绘制二维图形
plotyy	以不同 y 轴绘制二维图形
polar	绘制极坐标图形
print	实现当前图形的硬拷贝
semilogx	创建对数标度的 x 轴和线性标度的 y 轴
semilogy	创建对数标度的 y 轴和线性标度的 x 轴
set	设置对象属性
stairs	绘制梯形图
stem	绘制从 x 轴以直线连接离散数据点的图形
subplot	将当前图形分割为具有独立轴的若干子图
text	在当前轴添加文本对象
title	在当前轴添加标题
xlabel	标注当前 x 轴
ylabel	标注当前 y 轴
zoom	放大或缩小二维图形

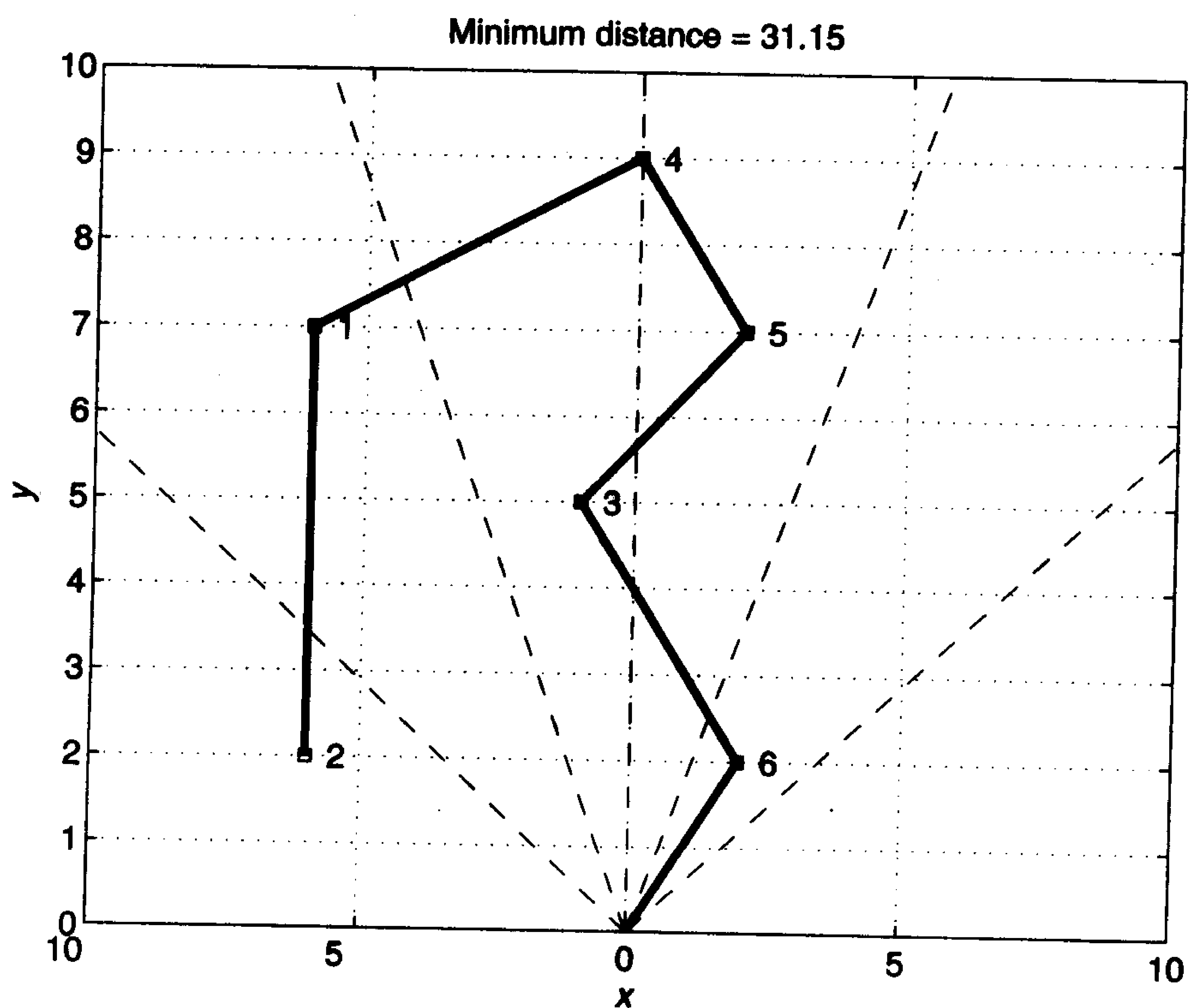


图 6.21 最小距离总和的 6 点排列次序

练习

注意:练习中图形的绘制可通过向量、点运算和 meshgrid 函数实现。for 循环仅用于提高参数的适用范围。

6.1 画出下面的曲线①,要求使用 axis equal 函数。

(a) 摆线($-\pi \leqslant \varphi \leqslant 3\pi$; $r_a = 0.5, 1, 1.5$):

$$x = r_a\varphi - \sin\varphi$$
$$y = r_a - \cos\varphi$$

(b) 双纽线($-\pi/4 \leqslant \varphi \leqslant \pi/4$):

$$x = \cos\varphi\sqrt{2\cos(2\varphi)}$$
$$y = \sin\varphi\sqrt{2\cos(2\varphi)}$$

(c) 螺旋线($0 \leqslant \varphi \leqslant 6\pi$):

i. 阿基米德曲线:

$$x = \varphi\cos\varphi$$
$$y = \varphi\sin\varphi$$

ii. 对数曲线($k = 0.1$):

$$x = e^{k\varphi}\cos\varphi$$
$$y = e^{k\varphi}\sin\varphi$$

(d) 心形线($0 \leqslant \varphi \leqslant 2\pi$):

$$x = 2\cos\varphi - \cos 2\varphi$$
$$y = 2\sin\varphi - \sin 2\varphi$$

(e) 星形线($0 \leqslant \varphi \leqslant 2\pi$):

$$x = 4\cos^3\varphi$$
$$y = 4\sin^3\varphi$$

(f) 外摆线($R_r = 3, a_r = 0.5$, 1 或 2, $0 \leqslant \varphi \leqslant 2\pi$; $R_r = 2.5, a_r =$ 2 且 $0 \leqslant \varphi \leqslant 4\pi$):

$$x = (R_r + 1)\cos\varphi - a_r\cos(\varphi(R_r + 1))$$
$$y = (R_r + 1)\sin\varphi - a_r\sin(\varphi(R_r + 1))$$

(g) 内摆线($R_r = 3$; $a_r = 0.5$, 1 或 2; $0 \leqslant \varphi \leqslant 2\pi$):

$$x = (R_r - 1)\cos\varphi + a_r\cos(\varphi(R_r - 1))$$
$$y = (R_r - 1)\sin\varphi - a_r\sin(\varphi(R_r - 1))$$

6.2 在 $-15 \leqslant x \leqslant 15$ 范围内绘制下列 Kilroy 曲线②:

$$y = \ln\left|\frac{\sin x}{x}\right|$$

---

① 参见 D. von Seggern, *CRC Standard Curves and Surfaces*, CRC Press, Inc., Boca Raton, FL, 1993 年。

② 参见 E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*,第二版,Chapman & Hall/CRC, Boca Raton, FL,2003 年,第 1625 页。

6.3 在 $-12 \leqslant x \leqslant 7$ 范围内，存在以下多项式：

$$y = 0.001x^5 + 0.01x^4 + 0.2x^3 + x^2 + 4x - 5$$

只画出正值。强迫 y 值在每段的起始点和终点等于或接近于 0。[提示:用 find 函数。]

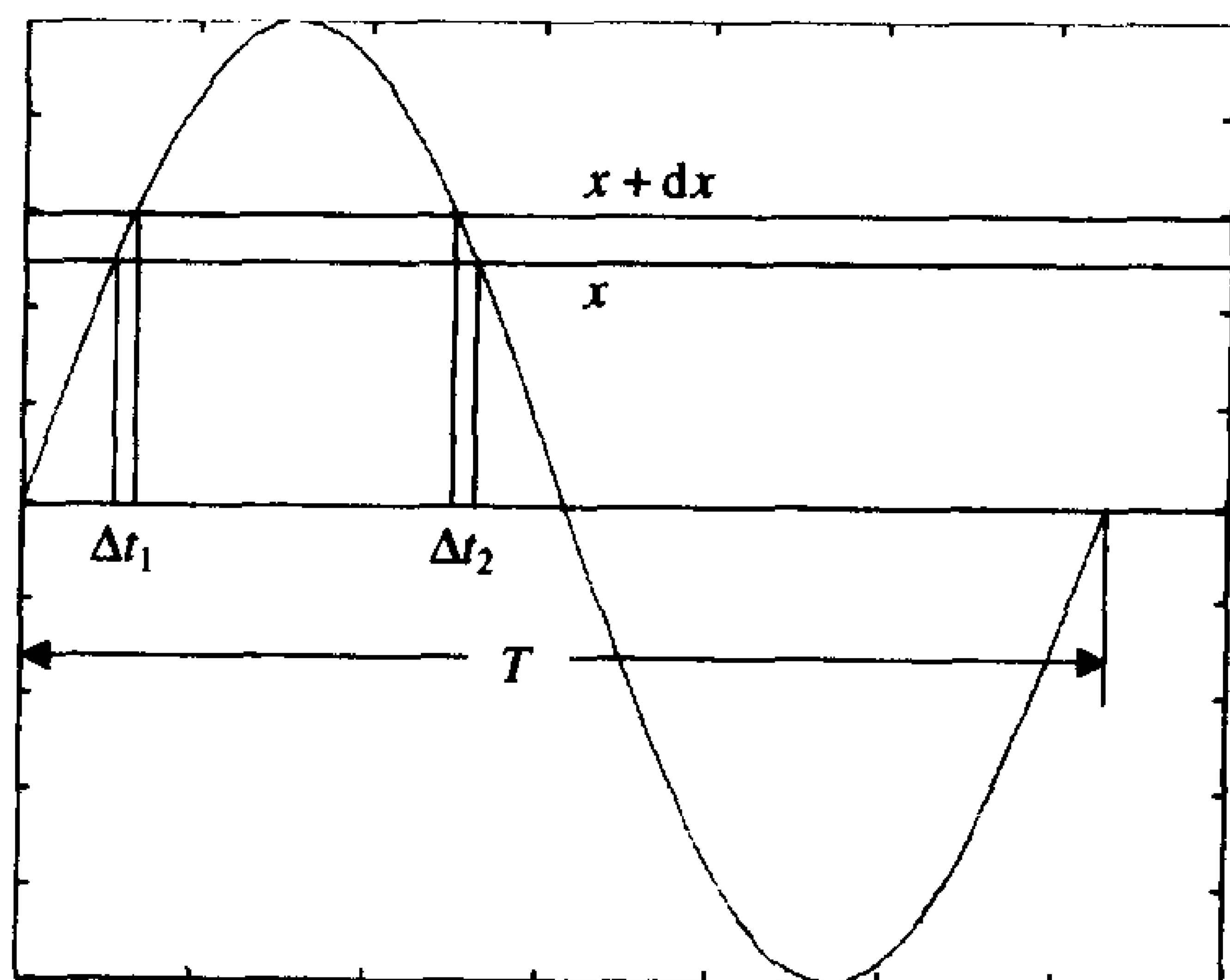


图 6.22　求出正弦波在 x 和 $x+dx$ 之间所经过的时间

6.4 时变信号的概率密度函数用于描述周期 T 内，信号振幅介于 x 和 $x+dx$ 之间的可能性。换言之，是用来得到幅值范围内信号时间的百分比。概率密度函数可近似为：

$$P(x) = \lim_{\substack{\Delta x \to 0 \\ T \to \infty}} \left[\frac{1}{T\Delta x} \sum_i \Delta t_i \right]$$

表达式中的各项在图 6.22 中标出，图 6.22 画出了正弦波的一个周期。振幅为 A_o 的正弦波的概率密度函数 $P(x)$ 如下：

$$P(x) = \frac{1}{\pi\sqrt{A_o^2 - x^2}} \qquad |x| \leqslant A_o$$

$$= 0 \qquad |x| > A_o$$

估算概率密度函数

$$y = A_0 \sin t$$

在 $-\pi \leqslant t \leqslant \pi$ 时的结果，并将其与实际值相比较。令 $A_o = 2$，振幅区间数等于 20，时间间隔内的数值点数为 400，画出 $P(x)$ 的估计值和实际值。结果如图 6.23 所示。

6.5 施加在贝莱维尔(Belleville)弹簧(参见图 8.45)上的力与 C_1 成比例，其中，

$$C_1 = 0.5d_t^3 - 1.5h_t d_t^2 + (1 + h_t^2)d_t$$

$h_t = h/t$，$d_t = \delta/t$，δ 为弹簧的挠度。当 h_t 在 1 到 3 之间变化，增量为 0.25 且 d_t 在 0 到 5 之间变化时，画出 C_1 随 d_t 变化的曲线，标注曲线并令 y 轴极限值为 8。结果应如图6.24所示。

6.6 齿轮齿如图 6.25 所示。如果齿轮有 n 个齿，则每个齿占 $2\pi/n$ 度。设 R_b 为基圆半径，R_T为齿节圆的半径，$R(R_b \leqslant R \leqslant R_T)$为齿轮廓上某点的半径。则轮齿轮廓极坐标为$(R, \phi)$，包括相邻齿间的空隙，如表 6.10 所示。表 6.10 中，ϕ_s 为齿轮压力角，值为 14.5°，20°或 25°，$R_s = nm/2$ 为标准齿节半径，m 为齿轮模数，t_s 为 R_s 处的齿轮厚度。如果齿轮有 24 齿，压力角为 20°，模数为 10 mm，齿厚为 14.022 mm，基圆半径为 90.21 mm，齿节圆半径为 106 mm，用两种方式 plot 和 polar 画出齿轮图形。

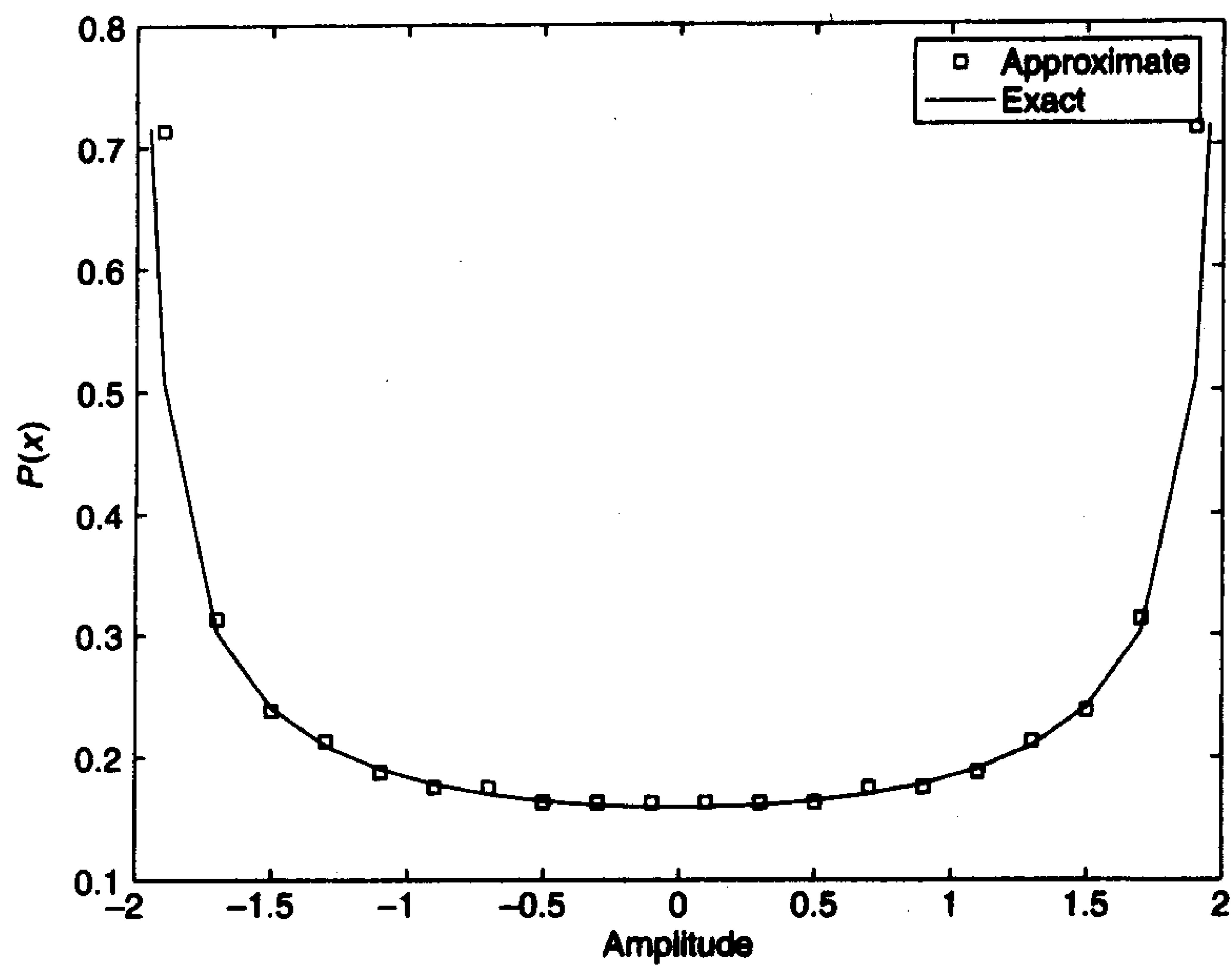


图 6.23 正弦波的概率密度函数

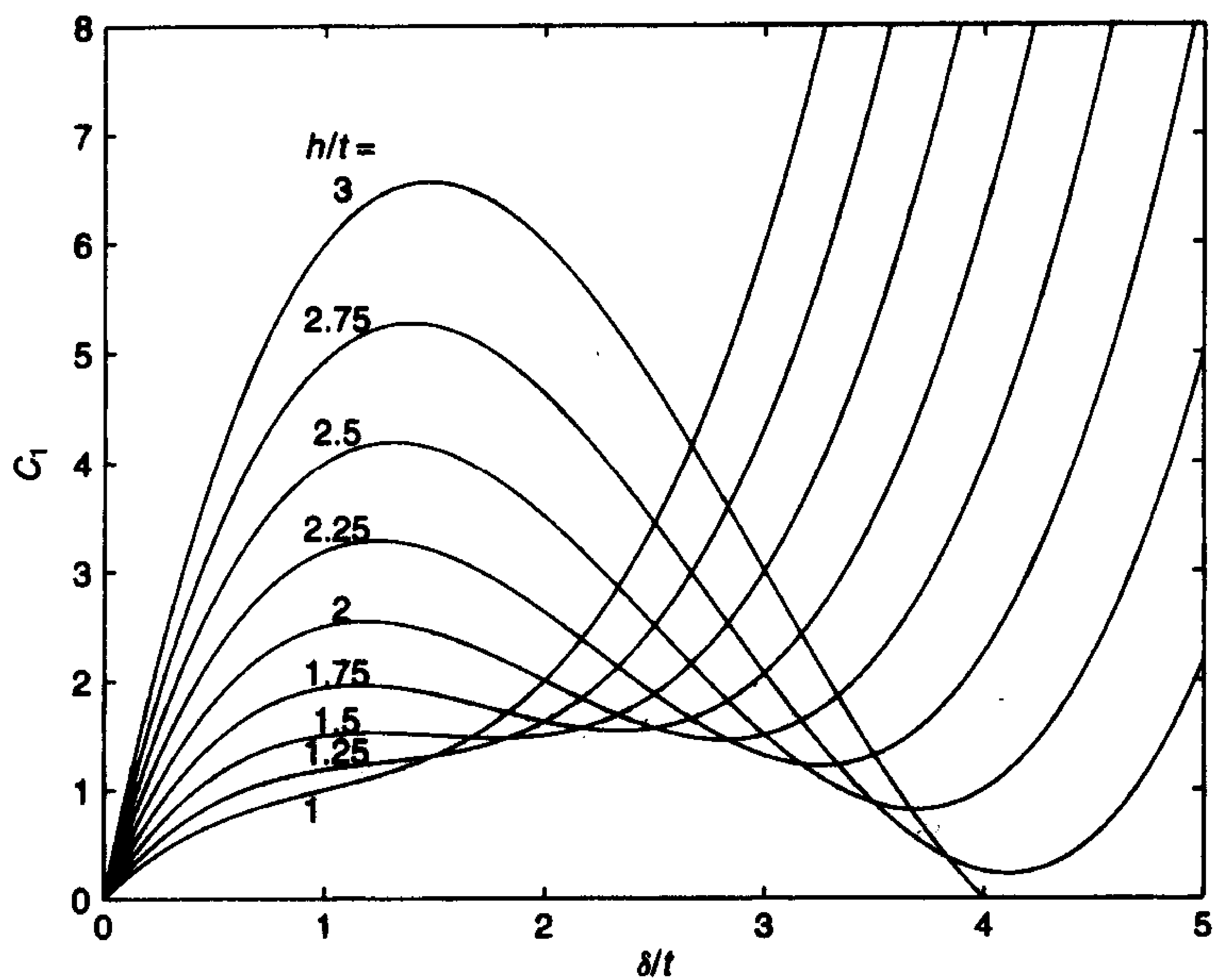


图 6.24 贝莱维尔弹簧的系数 C_1

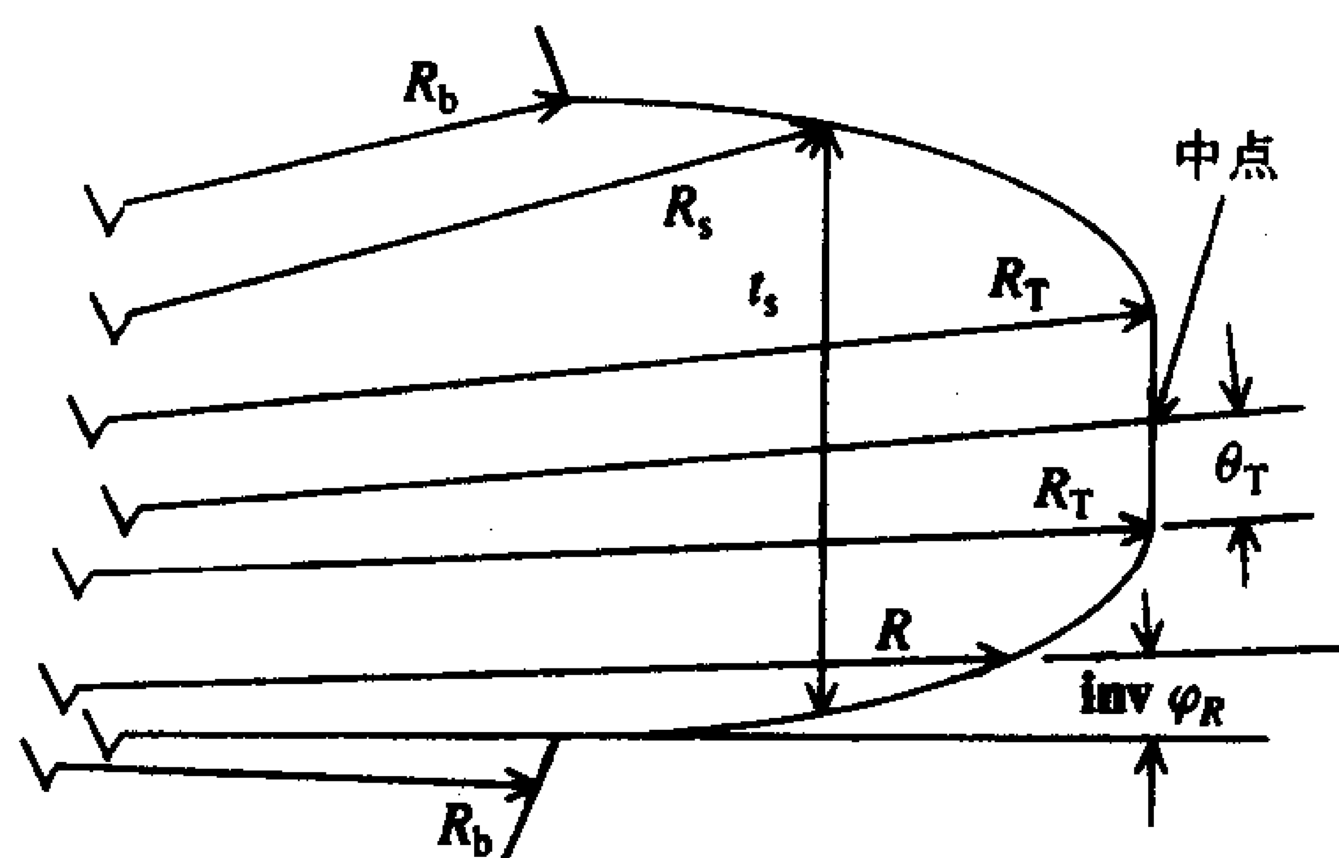


图 6.25 齿轮齿的几个参数

6.7 忽略轴环的摩擦,电钻效率的百分比为(参见习题8.11):

$$e = 100\frac{\cos(\alpha) - \mu\tan(\lambda)}{\cos(\alpha) + \mu\cot(\lambda)}\%$$

其中,μ 为摩擦系数,λ 为电钻的超前角,α 为螺纹角。当 $0 < \lambda < 90°$, $\mu = 0.02$, 0.05,0.10,0.15,0.20,0.25,螺纹角 $\alpha = 7°$和 14.5°时,画出效率随 λ 变化的函数曲线。标注图形及各条曲线,并用 axis 函数限制效率范围为 0 到 100%。结果应如图 6.26 所示。

表 6.10 图 6.25 所示齿轮齿的不同部位的意义

R	ψ	定义
$R_b \le R \le R_T$	$\mathrm{inv}(\varphi(R))$	$\varphi(R) = \cos^{-1}(R_b/R)$
		$\mathrm{inv}(x) = \tan(x) - x$
R_T	$\mathrm{inv}(\varphi(R_T)) \le \psi \le \mathrm{inv}(\varphi(R_T)) + 2\theta_T$	$\theta_T = 0.5t_s/R_s + \mathrm{inv}(\varphi_s) - \mathrm{inv}(\varphi(R_T))$
$R_b \le R \le R_T$	$2[\theta_T + \mathrm{inv}(\varphi(R_T))] - \mathrm{inv}(\varphi(R))$	
R_b	$2[\theta_T + \mathrm{inv}(\varphi(R_T))] \le \psi \le 2\pi/n$	

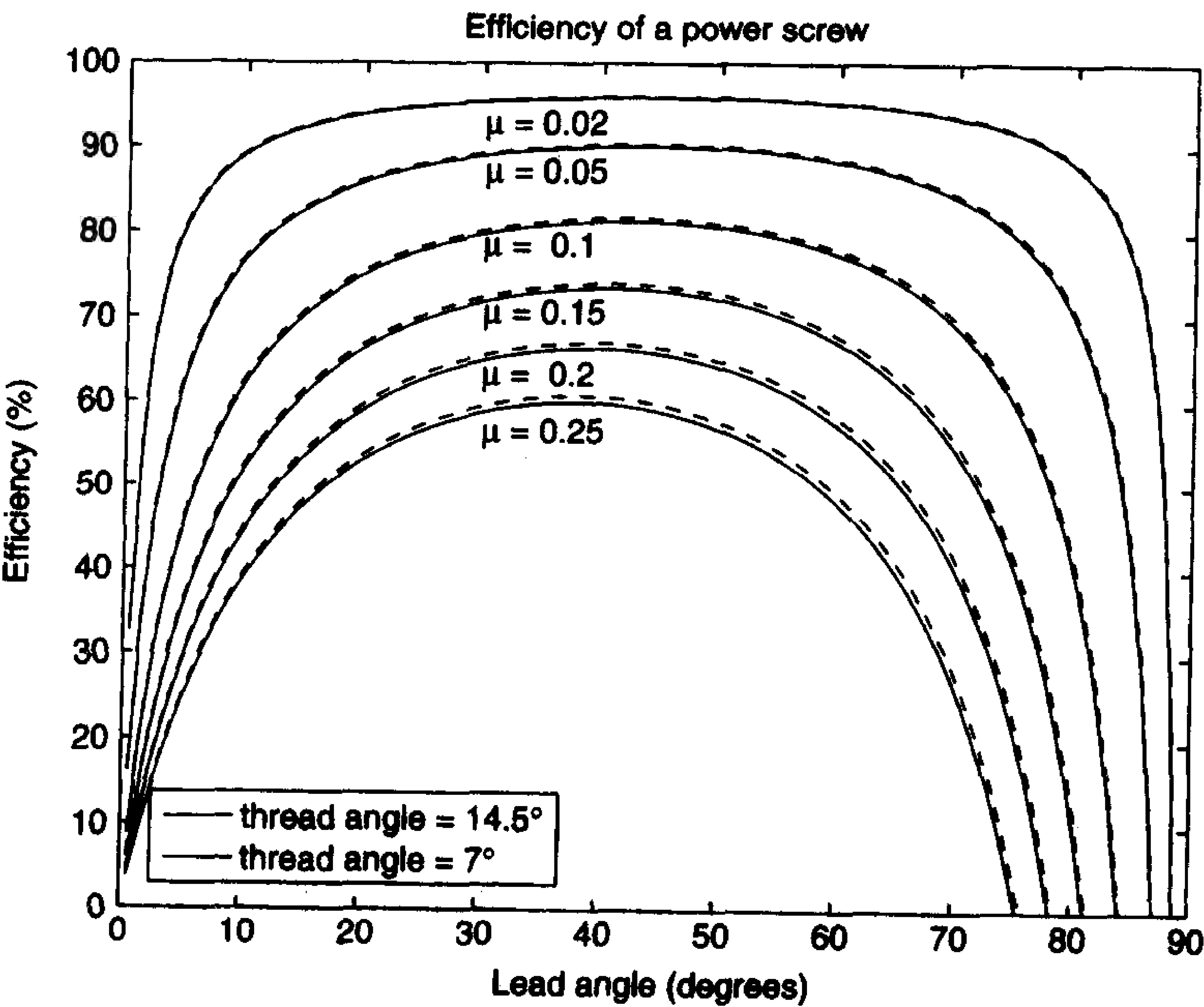


图 6.26 电钻的效率

6.8 矩形如图 6.27 所示,由图中看出:

$$r_1 = \sqrt{d^2 + (W/2)^2} \qquad \alpha = \tan^{-1}(W/2d)$$

$$r_2 = \sqrt{(d+L)^2 + (W/2)^2} \quad \beta = \tan^{-1}(W/2(d+L))$$

如果给出 L, W 和 d 的值,编写一段程序计算重复画在图 6.28 中不相交矩形的最大个数。其中,$L = 1$, $W = 2$, $d = 2$,可生成矩形的最大数目用 floor(π/α)确定。程序不使用 for 循环。

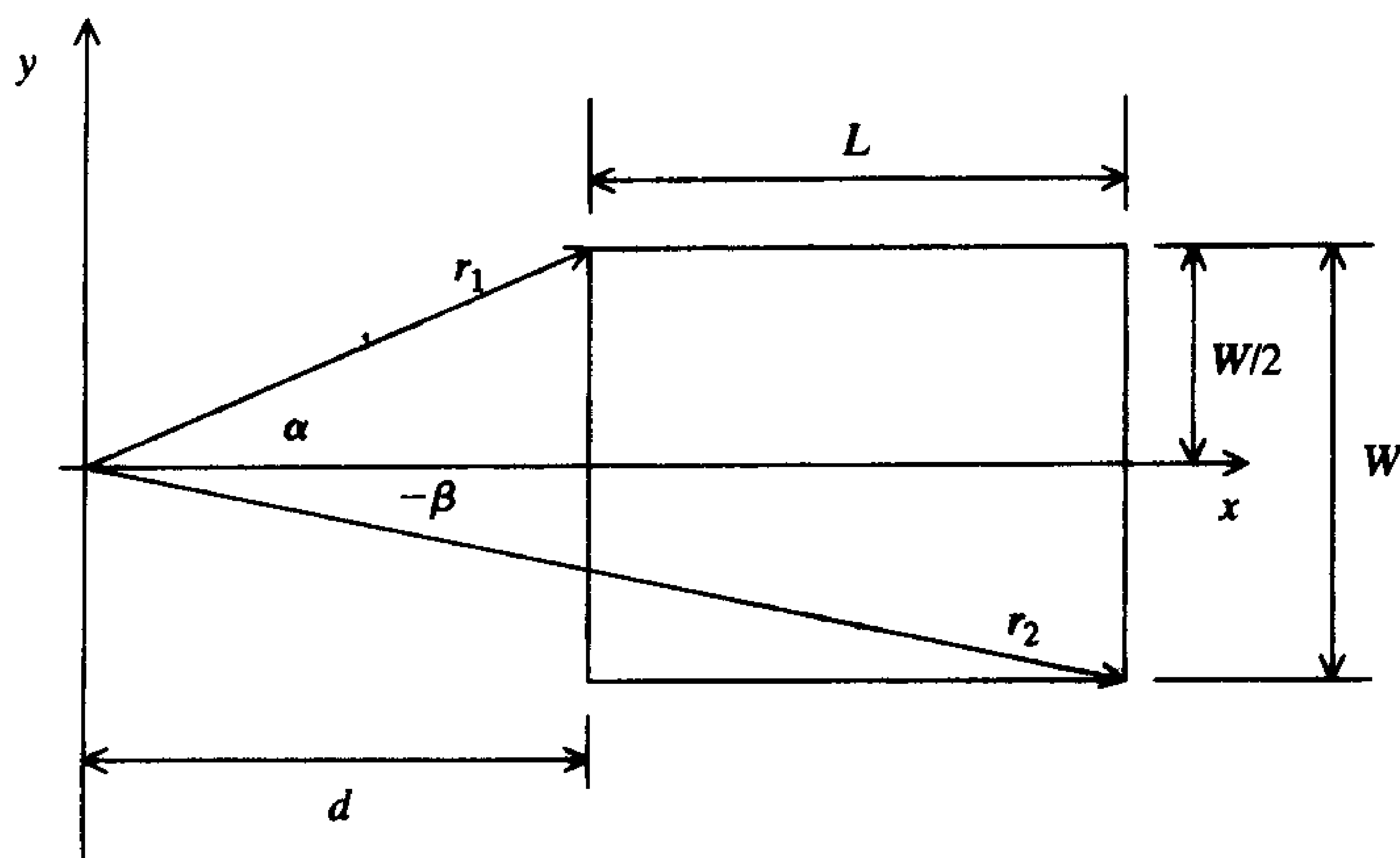


图 6.27　习题 6.8 中矩形的描述

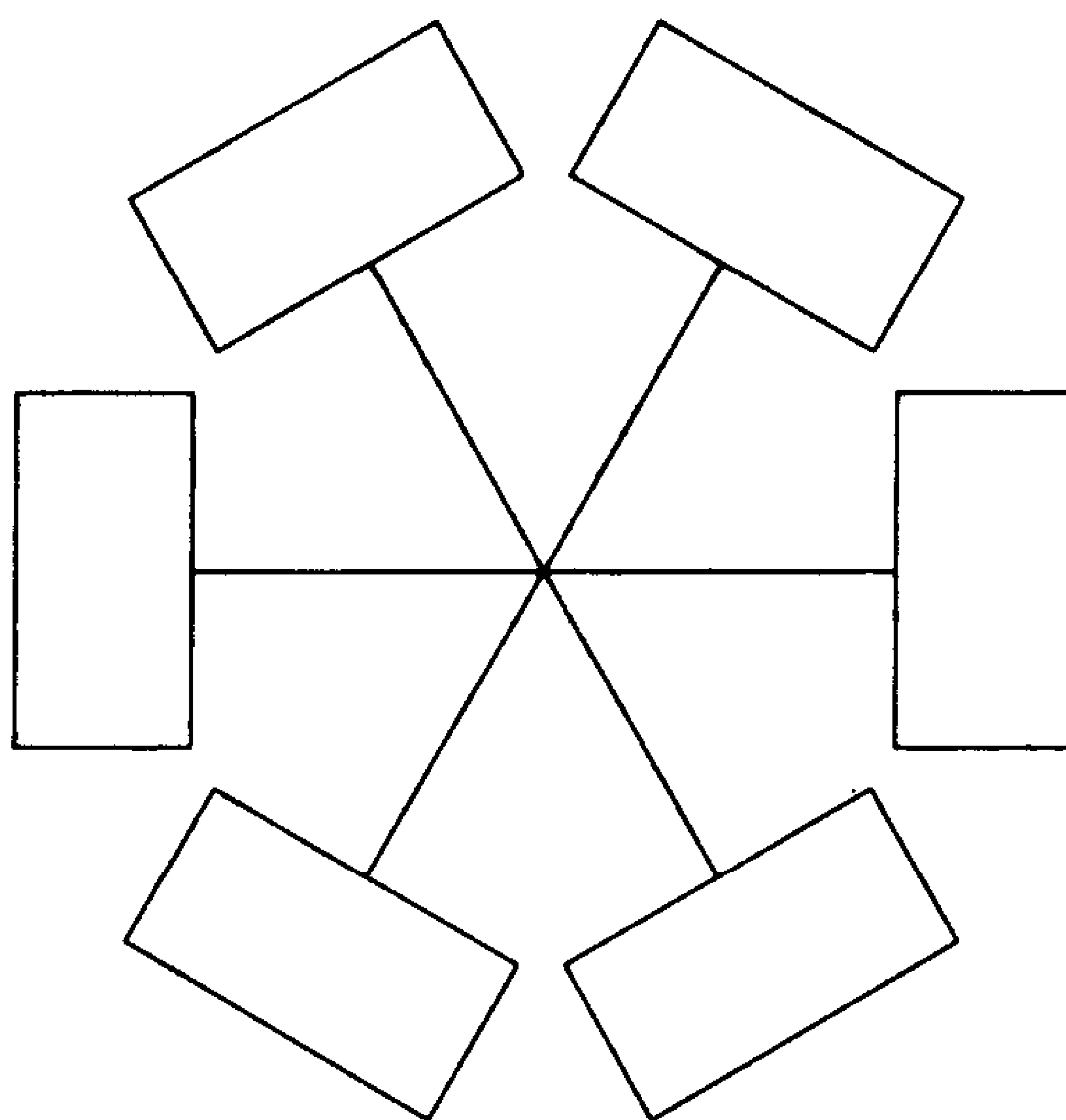


图 6.28　重复绘制的不相交矩形

6.9 使用习题 1.9 的结果，画出 $v_1 = 0.3$ 时，σ_x/p_{max}，σ_z/p_{max}，$\tau_{xz}/p_{max} = \tau_{yz}/p_{max} = 0.5(\sigma_x/p_{max} - \sigma_z/p_{max})$随 z/a 变化的图形。对图形和曲线加以标注。

6.10 使用习题 1.10 的结果，画出$\nu = 0.3$ 时，σ_x/p_{max}，σ_y/p_{max}，σ_z/p_{max}和 τ_{yz}/p_{max}随 z/b 变化的图形。对图形和曲线加以标注。

6.11 油的绝对黏度可由下式估算，单位为 μreyn(lb-s/in^2)，其误差为 ± 10%：

$$\mu = 10^{C-1}$$

其中，

$$C = 10^{A_j - B_j \log_{10} T_o}$$

$T_o = 255.2 + 5/9T$ K，其中 T 为油温，单位为°F，A_j，B_j 为油 SAE(简单算术表达式)数的函数，其值由表 6.11 给出。画出表 6.11 中 6 种油的 lglg(10μ)随lg(T_o)变化的函数关系曲线，以及 μ 随 T_o 变化的函数曲线(参见 8.6 节)。

表 6.11　确定 μ 的系数

SAE 数	j	A_j	B_j
10	1	9.1209	3.5605
20	2	9.1067	3.5385
30	3	8.9939	3.4777
40	4	8.9133	3.4292
50	5	8.5194	3.2621
60	6	8.3666	3.1884

6.12 蜗轮的超前角 λ;齿数比 $\beta = N_1/N_2$,其中 N_1 是蜗轮的齿数,N_2 是驱动轮的齿数;轴间中心距 C 和法向径节 P_{dn}之间的关系为:

$$K = \frac{2P_{dn}C}{N_2} = \frac{\beta}{\sin\lambda} + \frac{1}{\cos\lambda}$$

当 $1° \leqslant \lambda \leqslant 40°$,$\beta = 0.02, 0.05, 0.08, 0.11, 0.15, 0.18, 0.23, 0.30$ 时,画出 K 相对于 λ 的关系曲线。标注图形和曲线,y 轴范围限定在从 1 到 2 之间。在同一图形中用一条曲线连接每条曲线的最小值,从而得到习题 5.32 的结果。修改习题 5.32 中相关函数及部分程序,得到本题的程序。程序的执行结果应如图 6.29 所示。

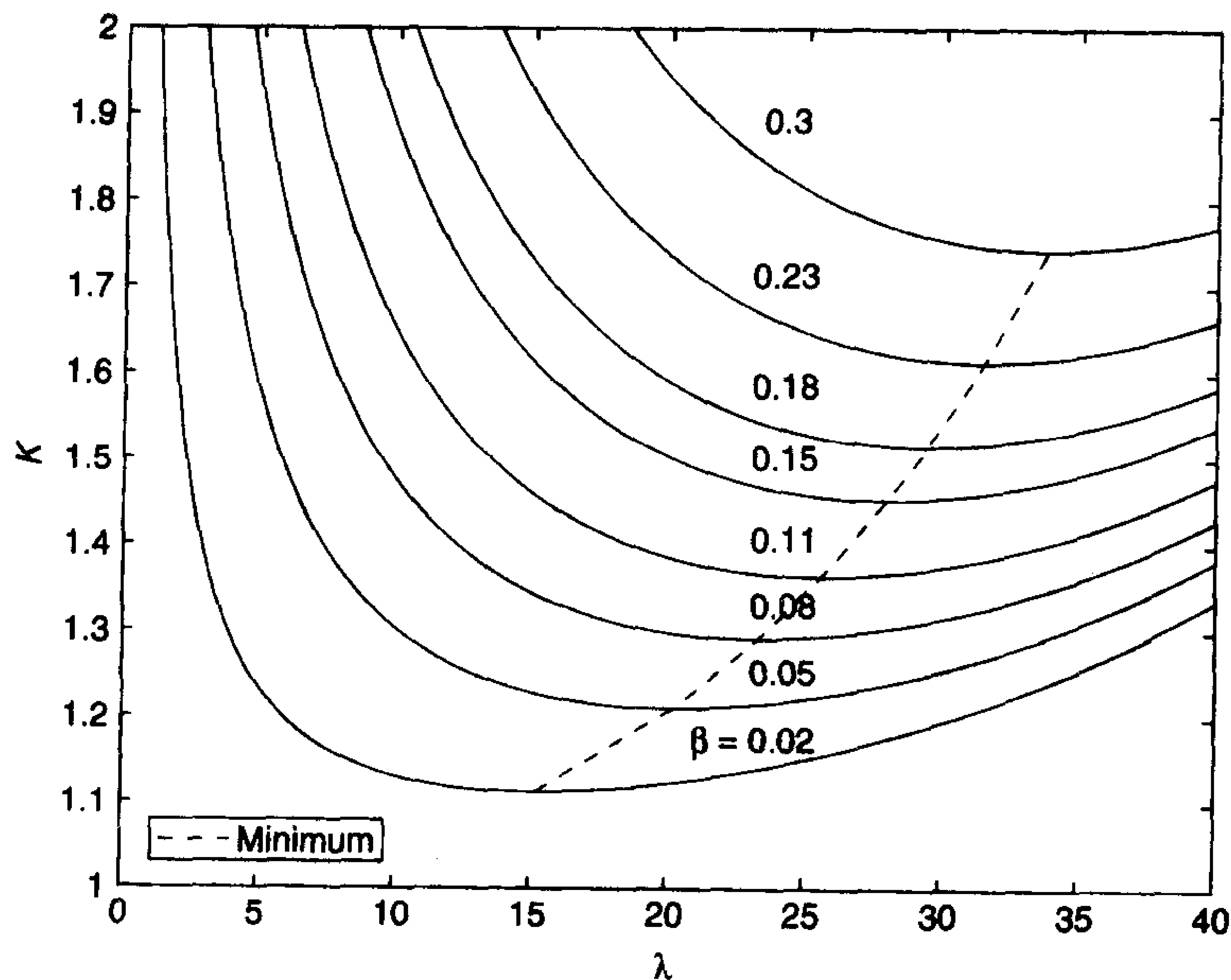


图 6.29　蜗轮的超前角

6.13 如图 6.30 所示,编写一段程序,使其在一个半径 $r_b = 1.5$ 的中心圆周围画出 3 个或 3 个以上的圆(图中是 5 个)。外部圆的半径 r_s 为:

$$r_s = \frac{r_b \sin(\pi/n)}{1 - \sin(\pi/n)}$$

程序要求用户输入需要画出的圆的个数,程序中未使用 for 循环。

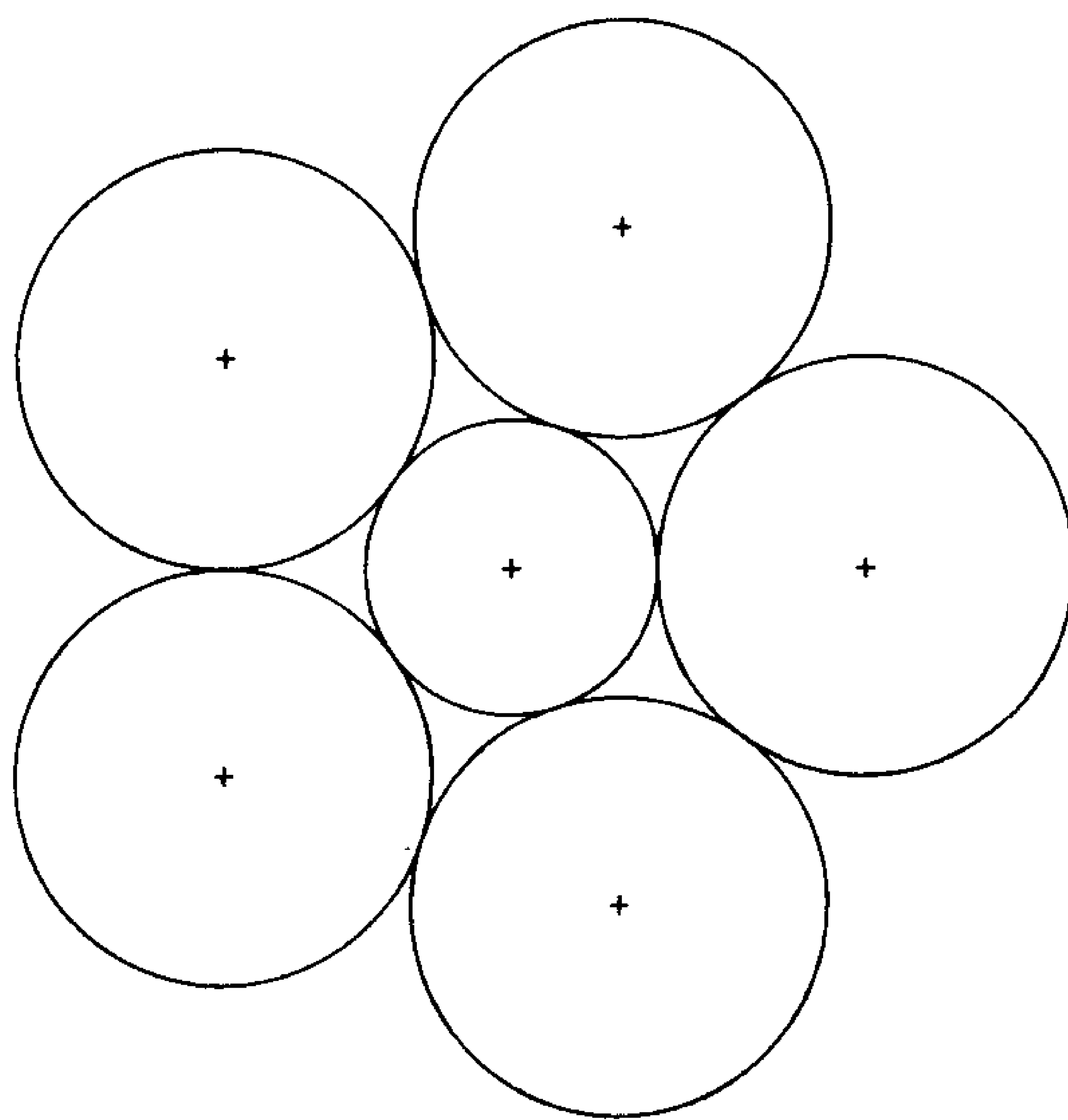


图 6.30 一个圆周围的 5 个圆

6.14 习题 5.19 中给出的柯尔勃洛克通用公式如下,用于估算管子的摩擦系数 λ:

$$\lambda = \left[-2\lg\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{0.27}{d/k}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

其中,R_e 为雷诺数,d 为管子的直径,k 为表面粗糙度。对于光滑管($k \cong 0$; $d/k >$ 100 000):

$$\lambda = \left[-2\lg\left(\frac{R_e\sqrt{\lambda}}{2.51}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

当 $4\times10^3 \leqslant R_e \leqslant 10^7$, d/k = 20, 50, 100, 200, 500, 1000, 2000, 5000, 10 000, 20 000, 50 000, 100 000 和 ∞ ($k=0$)时,画出 $\lg(\lambda)$ 随 $\lg(R_e)$ 变化的曲线。使用 `semilogx` 函数代替 `plot` 函数。标注图形和曲线。在 $R_e = 10^7$ 的右侧(即在图形右侧纵轴外侧)放置各曲线标识符。结果即为著名的管道流动摩擦系数的莫迪(Moody)图。结果应如图 6.31 所示。

6.15 在优化分析中画出被优化的函数(称为目标函数)及其约束条件(要求解值限定在给定区域内)是一种好的方法。考虑以下最小化问题:

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$

其约束条件为:

$$g_1 = (x_1 - 3)^2 + (x_2 - 1)^2 - 1 \leqslant 0$$

$$g_2 = 2x_1 - x_2 - 5 \leqslant 0$$

因此,解 x_{m1} 和 x_{m2} 一定在圆 $f(x_1, x_2)$ 上,并在 g_1, g_2 限定的区域内。

画出以上目标函数(圆)和解区间。结果应如图 6.32 所示。为此,应正确使用 `fill` 函数。

6.16 编写一段程序创建习题 5.21 的图 5.17,填充颜色为蓝绿色。

6.17 如图 6.33 所示,一个 N 对声源的直线排列,声源的振动频率为 ω,振幅为 Q。排列

中距离 r 处的总声压为:

$$P(r,\theta) = Z_o\left[\sum_{m=1}^{N}\left(1+\frac{md}{r}\cos\theta\right)\exp[j(-\varphi_m+mdk\cos\theta)]\right.$$
$$\left.+\sum_{m=1}^{N}\left(1-\frac{md}{r}\cos\theta\right)\exp[j(-\varphi_{-m}-mdk\cos\theta)]\right]$$

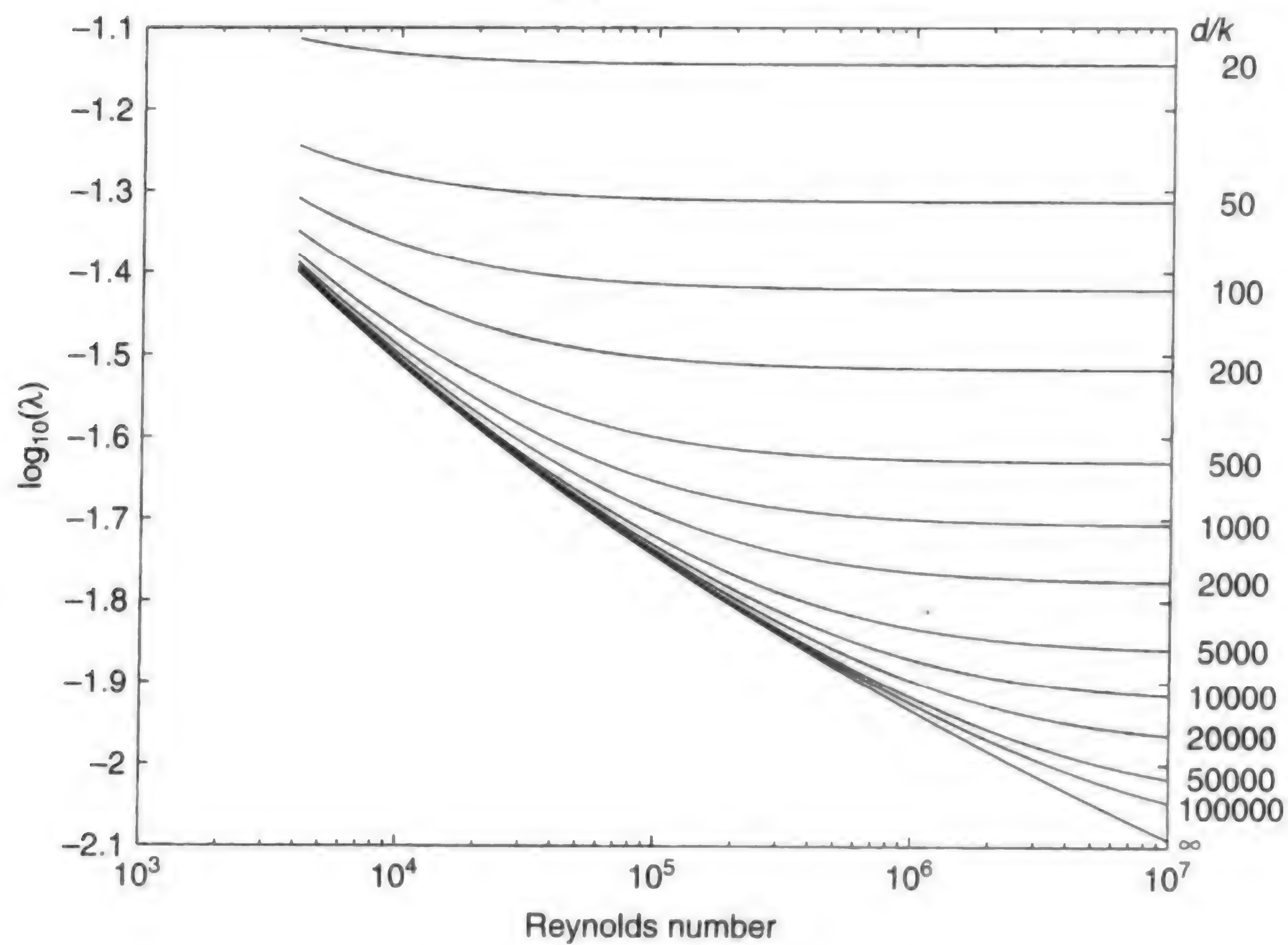


图6.31 莫迪图

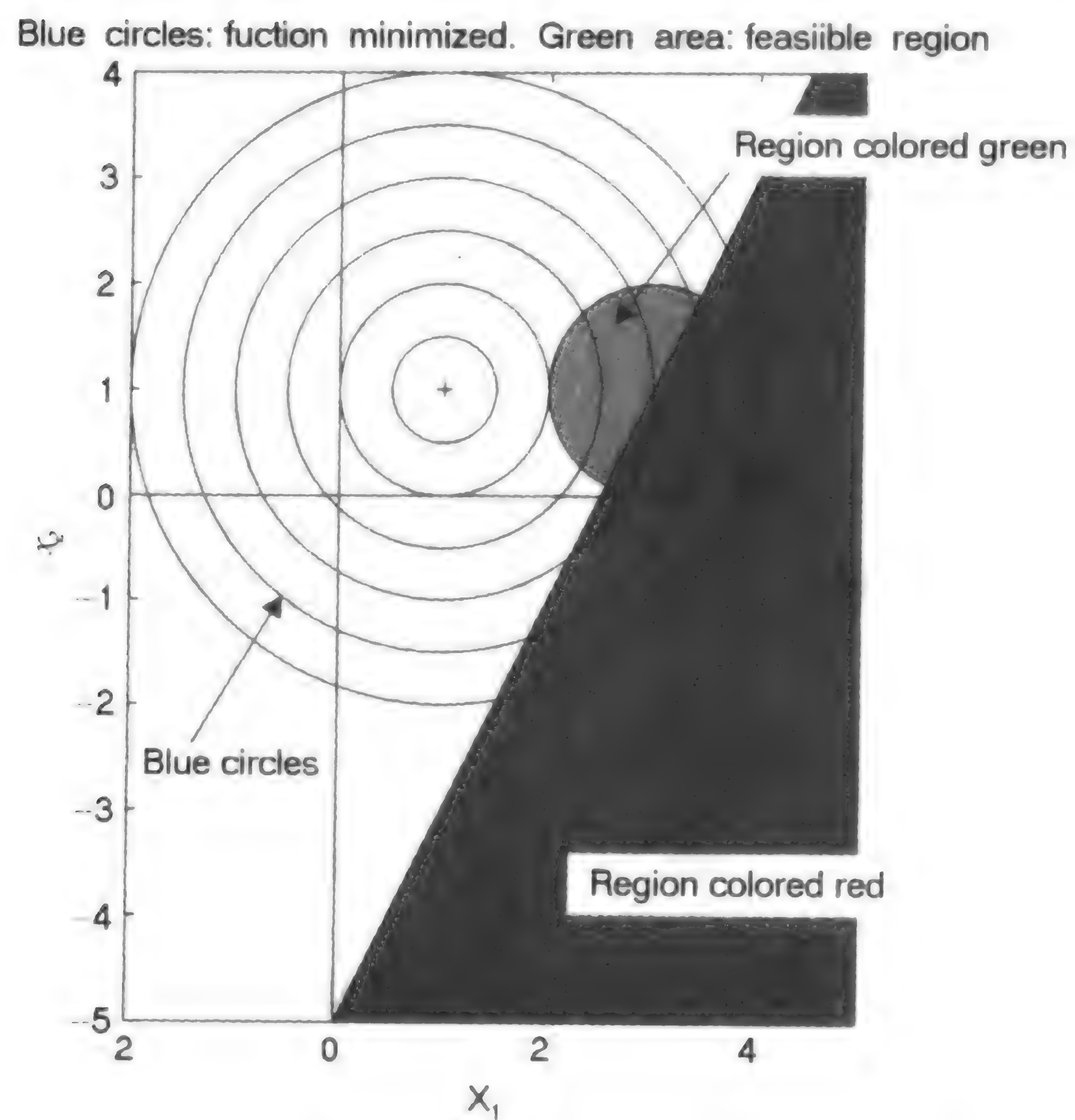


图6.32 习题6.15的解

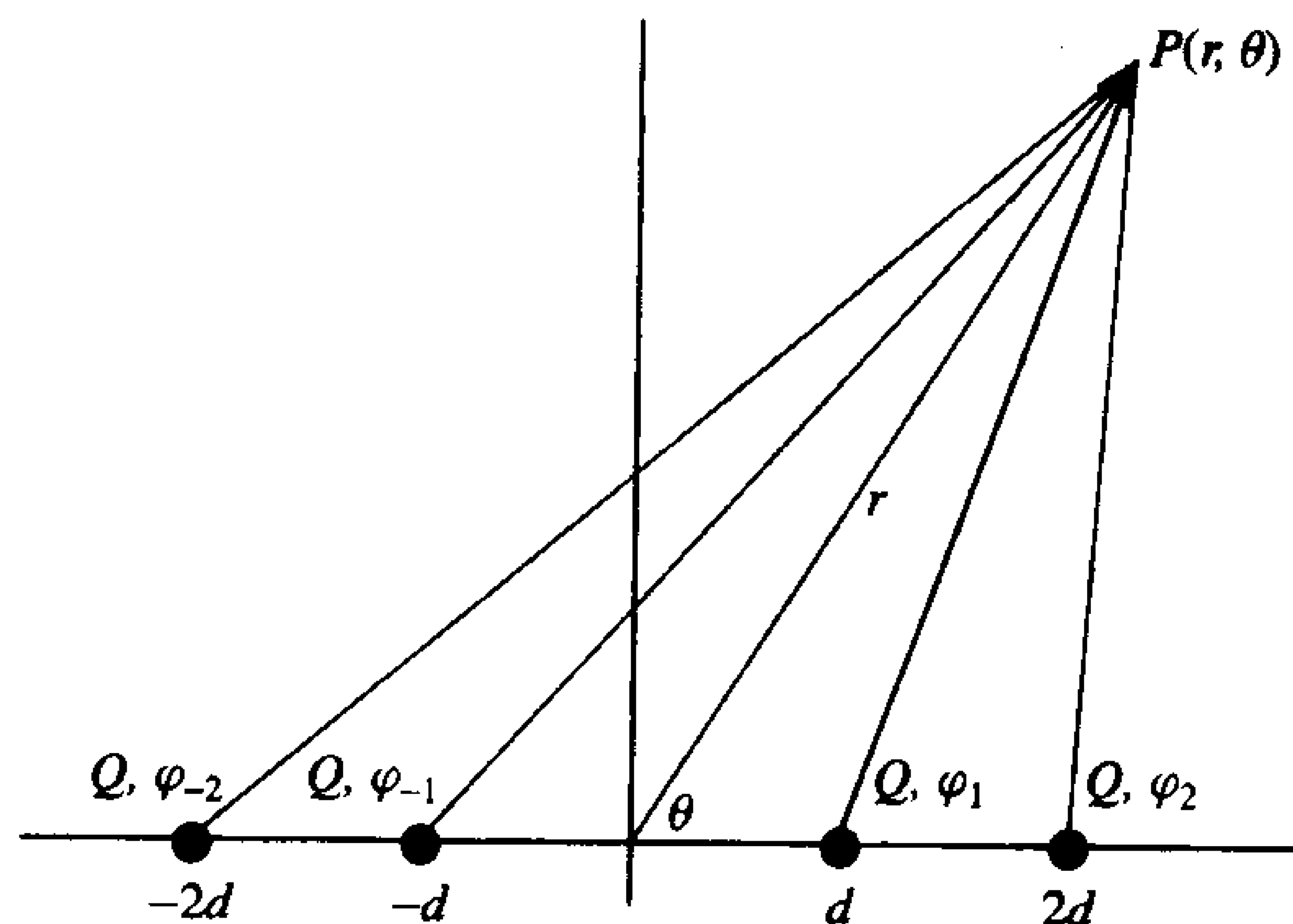


图 6.33 声源的线性排列

其中，

$$Z_o = \frac{j\rho ckQ}{4\pi r} e^{j(\omega t - kr)}$$

ρ 为介质密度，c 为在该介质中的音速，$k = 2\pi\omega / c$ 为声波数，$\phi_{\pm m}$ 为相对于 $\phi_1 = 0$，$d/r \ll 1$，$dk \ll 1$ 的相位角。

本例中 $N = 1$，$\phi_1 = 0$，$\phi_{-1} = \pi$，则上式可以简化为：

$$P(r, \theta) = 2Z_o\left[jdk + \frac{d}{r}\right]\cos\theta$$

该式为声音偶极子远域压力的表达式。

计算作为 θ 的函数 $|P(r,\theta)/Z_o|$，并当 $d/r = 0.05$，$dk = 0.03$，$\varphi_{-1} = \pi$，$\varphi_1 = 0$ 时，用 polar 函数画出 $0 \leqslant \theta \leqslant 2\pi$ 时的结果，如图 6.34 所示。用此公式分析直线排列的一般情况，同时画出偶极子精确解与数值分析解的相对误差。该图中不包含第一点($\theta = 0$)。首先，创建一个函数，计算不同声源条件下声压的幅值、相位关系和间隔以及空间中的远域位置。

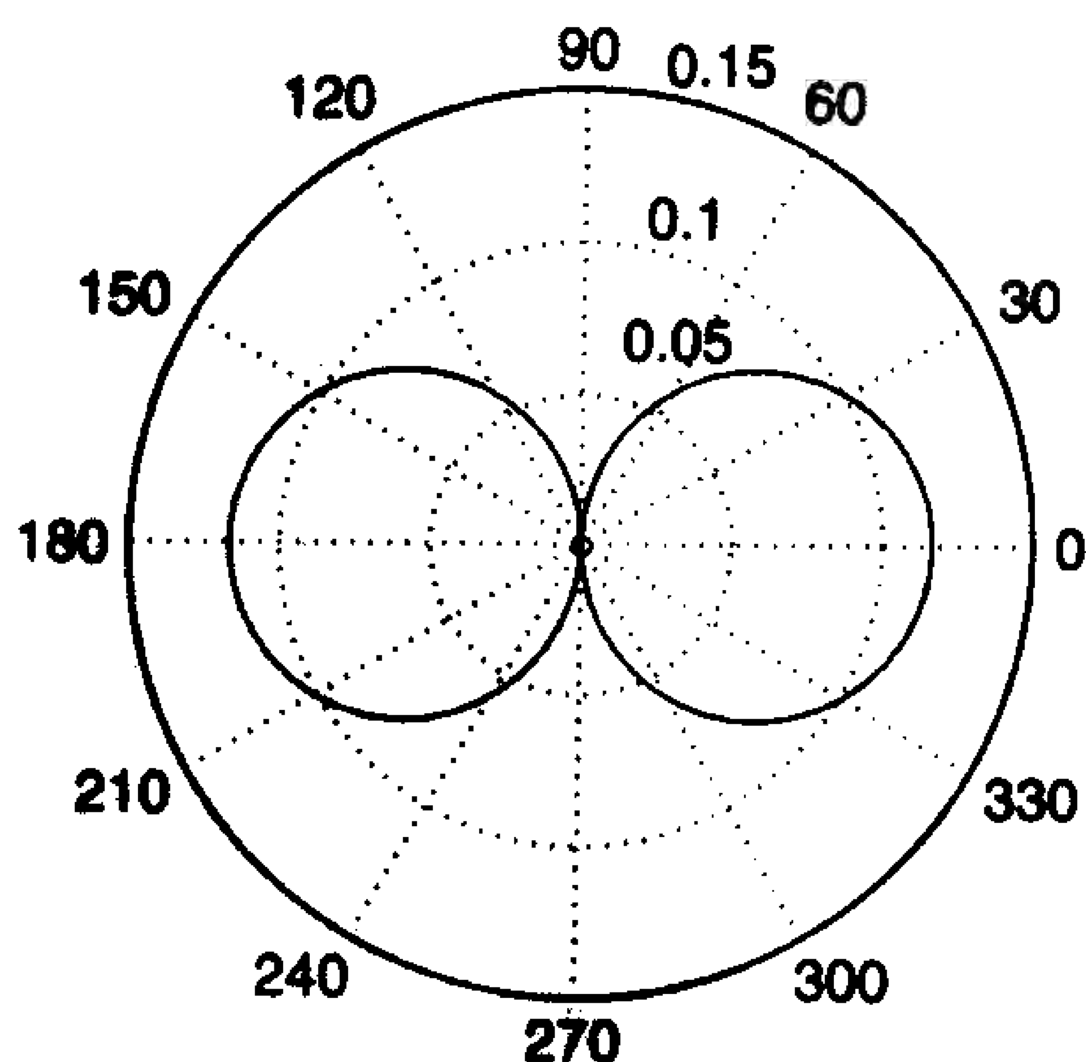


图 6.34 声音偶极子的辐射模式

第7章 三维图形

7.1 三维线

三维的 plot 函数为：

```
plot3(u1,v1,w1,c1,u2,v2,w2,c2,...)
```

其中，u_j，v_j 和 w_j 分别是某点的坐标。它们可以是标量、等长向量、等阶次矩阵，或者是在计算中可生成上述三种数值之一的表达式。c_j 是字符串，其中一个字符定义颜色，一个定义点的属性，最多两个字符定义线型(参见表6.1)。

假设要画一系列未连接的线，其端点为(x_{1j}, y_{1j}, z_{1j})和(x_{2j}, y_{2j}, z_{2j})，其中 $j = 1,2,\cdots,n$。为此，创建6个向量：

$$x_j = [x_{j1}\, x_{j2} \cdots x_{jn}]$$
$$y_j = [y_{j1}\, y_{j2} \cdots y_{jn}] \quad j = 1,2$$
$$z_j = [z_{j1}\, z_{j2} \cdots z_{jn}]$$

则 plot3 函数的指令为：

```
plot3([x1;x2],[y1;y2],[z1;z2])
```

其中，[x1;x2]，[y1, y2]和[z1, z2]均为$(2 \times n)$阶矩阵。使用 plot 函数画出一个三维相似形。

除了 text 函数的参数变为：

```
text(x,y,z,s)
```

其中，s 为字符串，且

```
zlabel
```

用于标注 z 轴外，其他所有在6.3节讨论过的二维画图的注释过程均适用于三维曲线及曲面的生成函数。

例7.1　线框盒

如图7.1所示，盒子的尺寸为 $L_x \times L_y \times L_z$。创建函数 **BoxPlot3**，画出盒子6个面的4个边。并利用该函数画出几个盒子。盒子的位置和方向由其两个对角的坐标 $\mathbf{P}(x_o, y_o, z_o)$ 和 $\mathbf{P}(x_o \pm L_x, y_o \pm L_y, z_o \pm L_z)$确定：

```
function BoxPlot3(x0,y0,z0,Lx,Ly,Lz)
x=[x0   x0      x0      x0      x0+Lx  x0+Lx  x0+Lx  x0+Lx];
y=[y0   y0      y0+Ly   y0+Ly   y0     y0     y0+Ly  y0+Ly];
z=[z0   z0+Lz   z0+Lz   z0      z0     z0+Lz  z0+Lz  z0];
index=zeros(6,5);
index(1,:)=[1  2  3  4  1];
```

```
index(2,:)=[5  6  7  8  5];
index(3,:)=[1  2  6  5  1];
index(4,:)=[4  3  7  8  4];
index(5,:)=[2  6  7  3  2];
index(6,:)=[1  5  8  4  1];
for k=1:6
  plot3 (x(index(k,:)),y(index(k,:)),z(index(k,:)))
  hold on
end
```

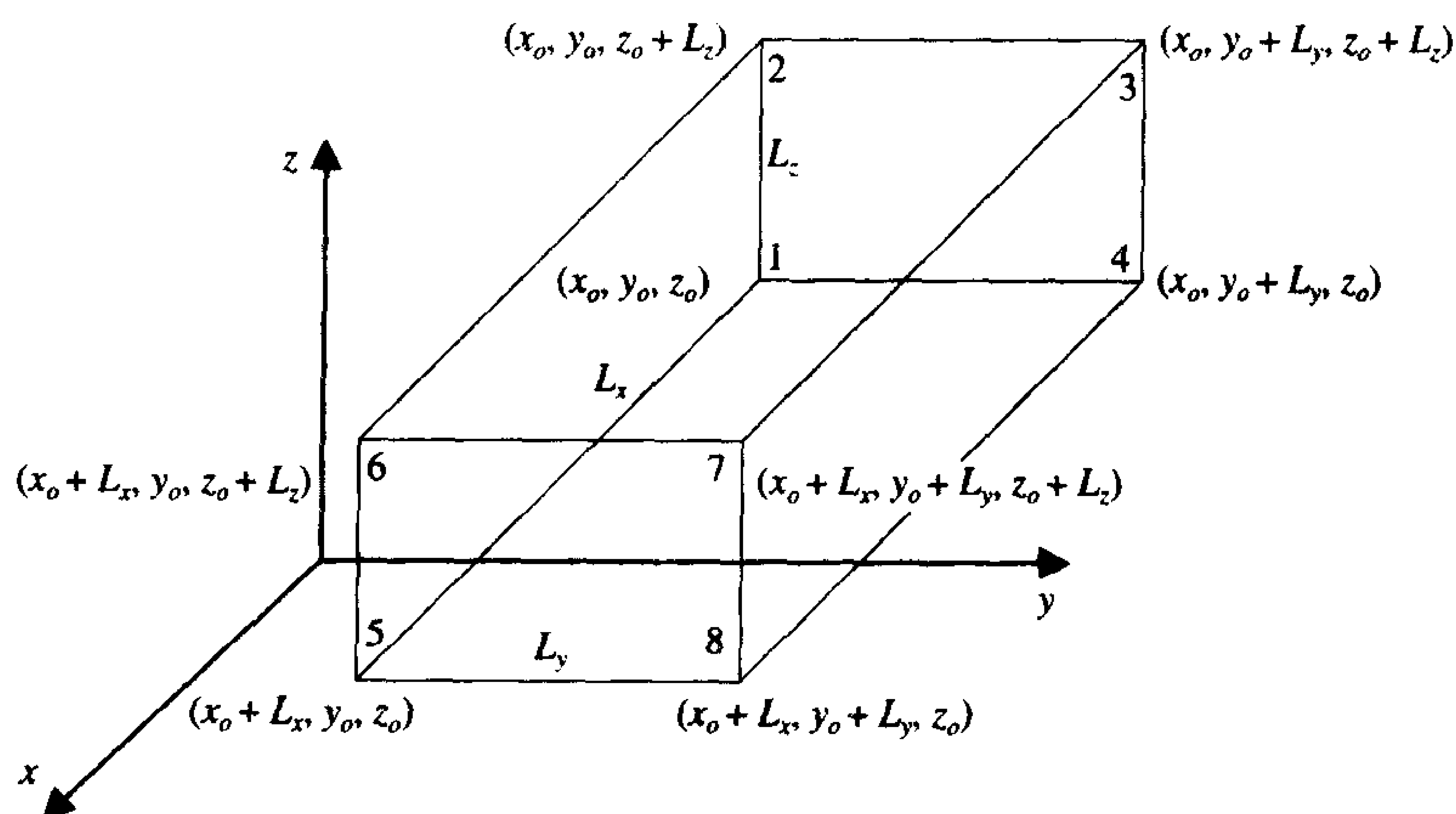


图 7.1　盒的坐标

以下程序创建了三个盒体,并给出其三维尺寸和一个角的坐标(x_o,y_o,z_o):

盒体 #1:

尺寸:3×5×7

坐标:(1,1,1)

盒体 #2:

尺寸:4×5×1

坐标:(3,4,5)

盒体 #3:

尺寸:1×1×1

坐标:(4.5,5.5,6)

创建和显示这些线框盒的程序如下:

BoxPlot3(1,1,1,3,5,7)

BoxPlot3(4,6,8,4,5,1)

BoxPlot3(8,11,9,1,1,1)

结果如图 7.2 所示。

例 7.2　在圆柱表面绘制正弦曲线

圆柱表明正弦曲线的坐标由下式获得①:

① 参见 D. von Seggern, *CRC Standard Curves and Surfaces*, CRC Press, Inc., Boca Raton, FL, 1993 年。

$$x = b\ \cos(t)$$
$$y = b\ \sin(t)$$
$$z = c\ \cos(at)$$

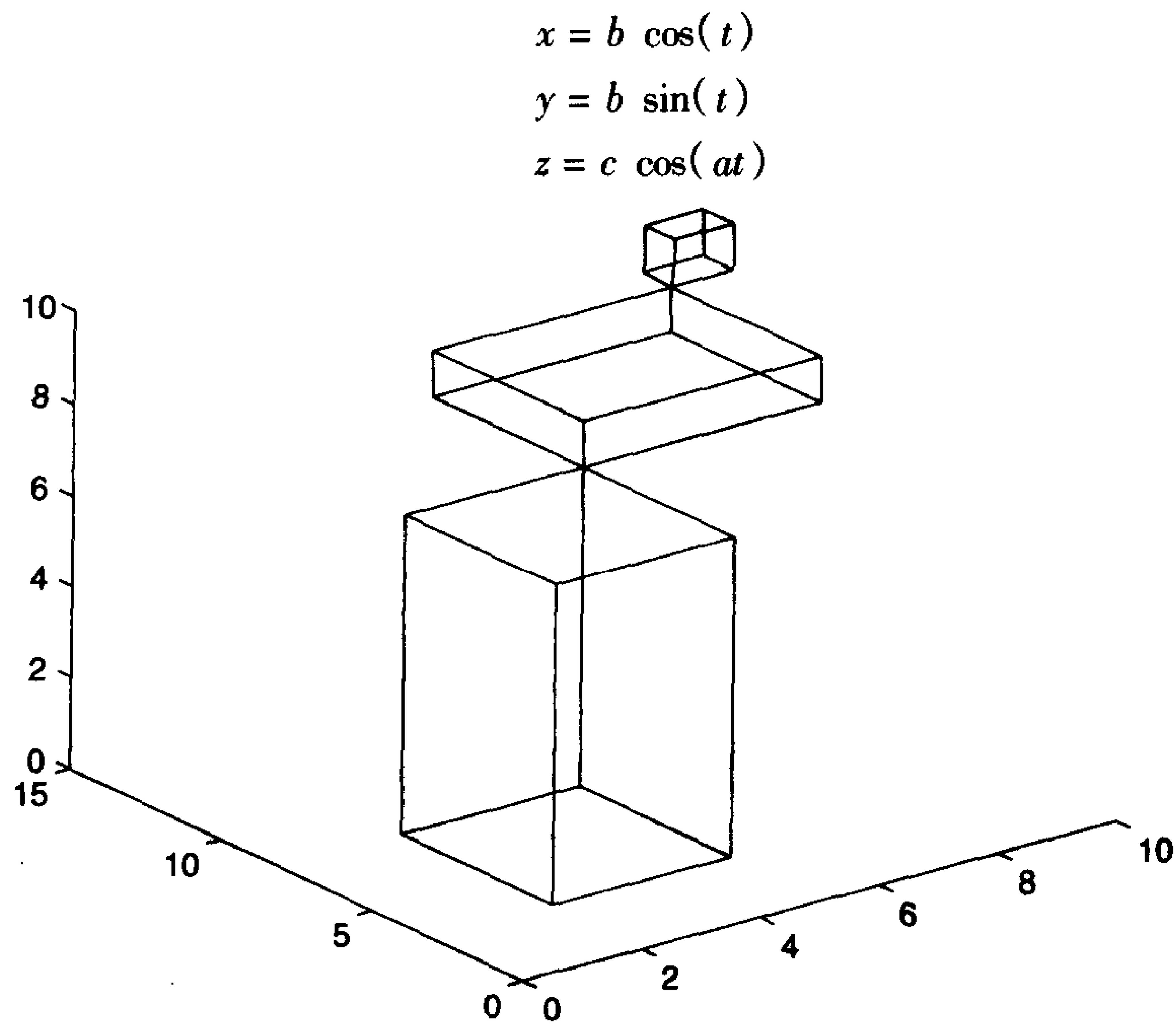


图 7.2 3个线框盒

令 $a = 10.0, b = 1.0, c = 0.3$,且 $0 \leqslant t \leqslant 2\pi$。则程序为:

```
t = linspace(0,2 * pi,200);
a = 10; b = 1.0; c = 0.3;
x = b * cos(t);
y = b * sin(t);
z = c * cos(a * t);
plot3(x,y,z,'k')
axis equal
```

执行结果如图 7.3 所示。

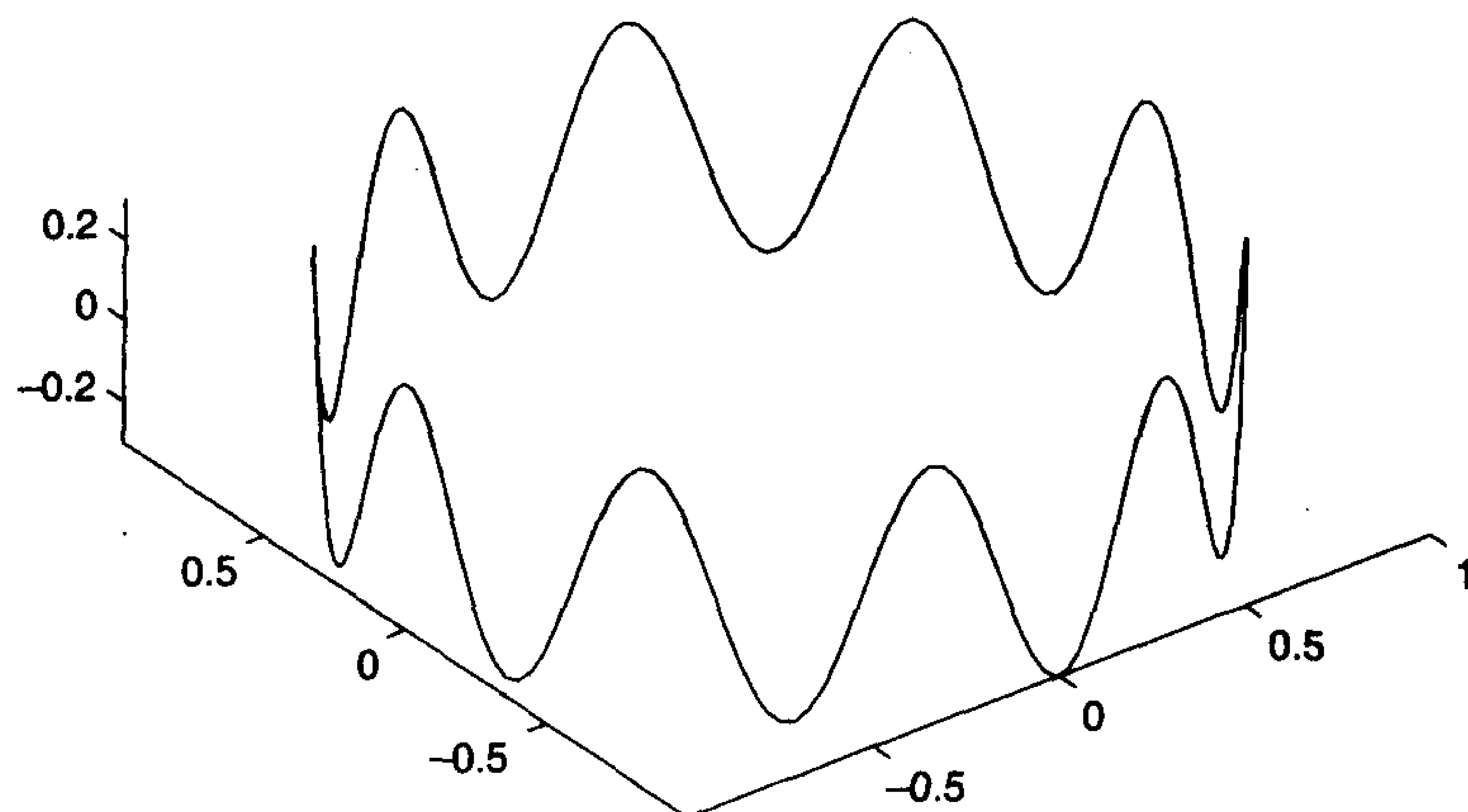


图 7.3 圆柱表面的正弦曲线

7.2 曲面

三维画图函数可生成曲面图、等高线图以及这些基本图形的变形和特殊化处理。曲面的定义表达式为:

$$z = f(x, y)$$

其中,x,y 是 xy 平面的坐标,z 是得出的高度。基本的曲面绘图函数为:

```
surf(x,y,z)
```

和

```
mesh(x,y,z)
```

其中,x,y 和 z 为曲面上点的坐标。函数 surf 画出的曲面由彩块组成,而 mesh 画出由边界给定的白色曲面。在 surf 函数中,块的颜色由 z 轴的值决定,而 mesh 中线的颜色也由 z 轴数值决定。

下面将在创建下式所确定的三维曲面过程中进一步介绍以上两个函数和其他几个函数的用法。

$$z(x, y) = x^4 + 3x^2 + y^2 - 2x - 2y - 2x^2y + 6$$

其范围为 $-3 < x < 3$, $-3 < y < 13$。为便于在其他例子中调用,创建函数名为 **SurfExample** 的函数文件生成 x,y 和 z 的坐标值:

```
function[x, y, z] = SurfExample
x1 = linspace( - 3,3,15);
y1 = linspace( - 3,13,17);
[x,y] = meshgrid(x1,y1);
z = x.^4 + 3 * x.^2 - 2 * x + 6 - 2 * y. * x.^2 + y.^2 - 2 * y;
```

表 7.1 给出了执行函数 **SurfExample** 后用 mesh 函数和 surf 函数所绘制的曲面的不同之处。

同时绘制曲面和曲线

为说明 3D 绘图函数可同时绘制多个曲面和多条曲线,创建以下两个函数。第 1 个函数为 **Corners**,从 **SurfExample** 函数创建曲面的 xy 平面的四个角,分别画直线到 $z = 0$ 平面,这些角的顶点坐标为:$(-3, -3, z(-3, -3))$, $(-3, 13, z(-3, 13))$, $(3, 13, z(3, 13))$ 和 $(3, -3, z(3, -3))$。第 2 个函数为 **Disc**,创建圆心为(0, 5),半径为 10 的圆盘并与曲面相交于 $z_o = 80$。绘制曲面、圆盘和连接到曲面四角的直线的程序为:

```
function SurfDisc
[x,y,z] = SurfExample;
surf (x,y,z)
Disc (10,80)
Corners

function Corners
xc = [ - 3, - 3,3,3];
```

```
yc = [ - 3,13,13, - 3];
zc = xc.^4 + 3 * xc.^2 - 2 * xc + 6 - 2 * yc. * xc.^2 + yc.^2 - 2 * yc;
hold on
plot3([xc;xc],[yc;yc],[zeros(1,4);zc],'k')

function Disc(R,zo)
r = linspace (0,R,12);
theta = linspace(0,2 * pi,50);
x = cos(theta') * r;
y = 5 + sin(theta') * r;
hold on
z = repmat(zo,size(x));
surf(x,y,z)
```

表 7.1　surf 函数与 mesh 函数差异的说明

绘图函数	程序	图　形
surf	[x,y,z] = SurfExample; surf(x,y,z)	200, 150, 100, 50, 0, 15, 10, 5, 0, -5, -4, -2, 0, 2, 4
mesh	[x,y,z] = SurfExample; mesh(x,y,z)	200, 150, 100, 50, 0, 15, 10, 5, 0, -5, -4, -2, 0, 2, 4

程序执行结果如图 7.4 所示。第四条线被图形遮住。

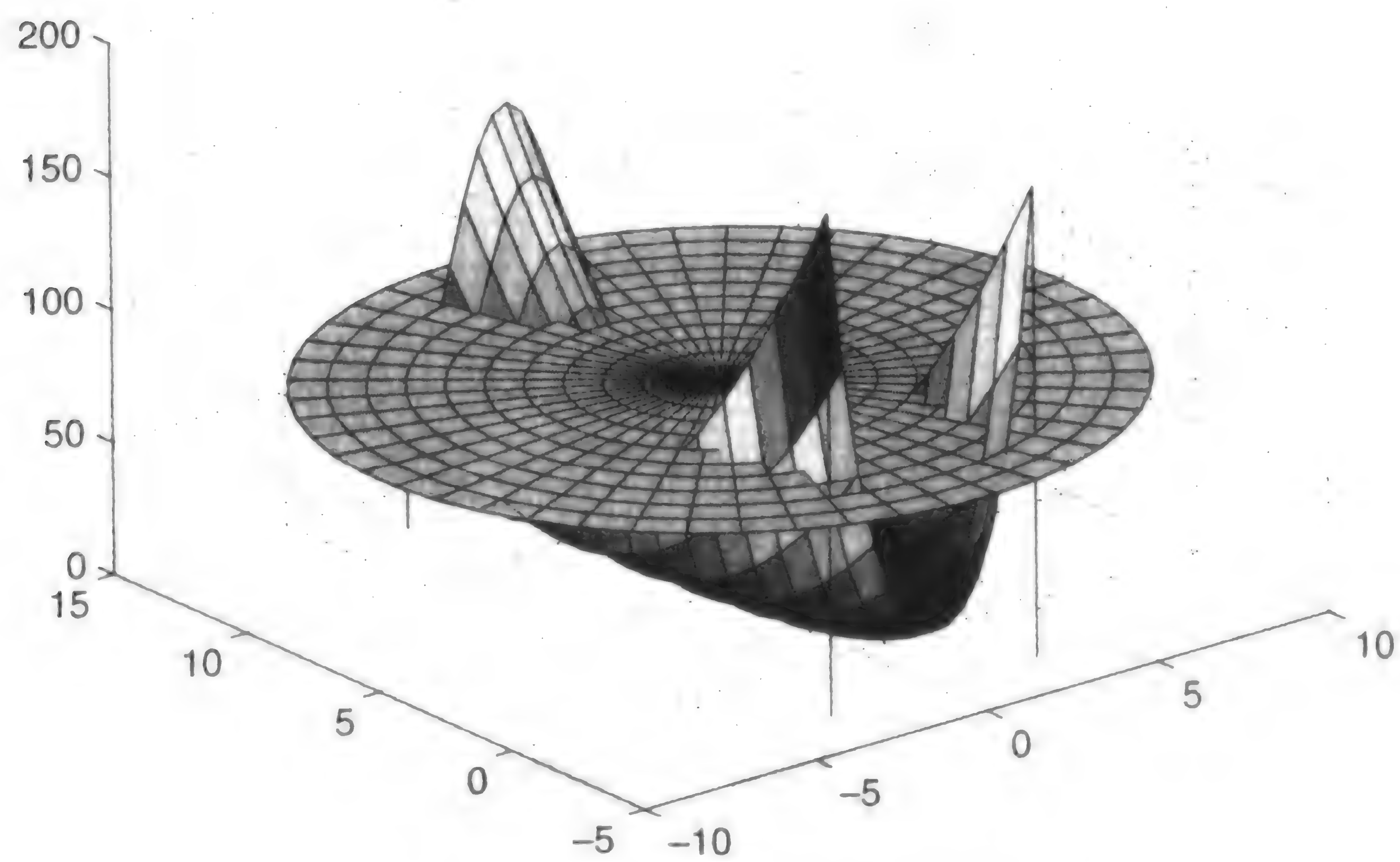


图 7.4 绘制四角直线及相交圆盘的曲面

改变图形外观

下列函数可任意组合,用来改变绘制曲面的外观:

```
box  on  或  box  off
grid  on  或  grid  off
axis  on  或  axis  off
```

axis on 函数有效时,函数 box on 绘制一个盒子。表 7.2 给出了这些函数组合的一些示例。

表 7.2 函数 box,grid 和 axis 的说明

绘图函数	程序	图 形
gird off	[x,y,z] = **SurfExample**; mesh(x,y,z) grid off	

（续表）

绘图函数	程序	图　形
axis off grid off	[x,y,z] = **SurfExample**; mesh(x,y,z) axis off grid off	
box on axis off grid off	[x,y,z] = **SurfExample**; mesh (x,y,z) box on axis on grid off	200 150 100 50 0 15 10 5 0 −5 −4 −2 0 2 4

如前所述，surf 函数绘制的图形方块颜色由 z 轴方向的值确定。与之类似，mesh 函数绘制的曲线颜色也由其 z 轴数值确定。但其颜色都可通过函数

```
colormap(c)
```

改变。其中，c 是一个在 0 和 1 之间变化的三元向量。第一个元素代表红色强度，第二个元素代表绿色强度，第三个元素代表蓝色强度。常用颜色的组合值参见表 7.3。

表 7.3　colormap(c)中颜色向量的常用值

c	颜色
[0 0 0]	黑
[1 1 1]	白
[1 0 0]	红
[0 1 0]	绿
[0 0 1]	蓝
[1 1 0]	黄
[1 0 1]	深红
[0 1 1]	青色
[0.5 0.5 0.5]	灰色

表 7.4 给出了另外两种加强视觉效果的方法，用于由函数 **SurfExample** 创建的曲面。

表 7.4　meshz 和 waterfall 函数说明

绘图函数	程序	图　形
meshz	[x,y,z] = SurfExample; meshz(x,y,z)	
waterfall	[x,y,z] = SurfExample; waterfall(x,y,z)	

等高线图

通过曲面可绘制各种等高线图,即根据给定的 z 值绘制的平行于 xy 平面的平面与曲面相交的曲线。函数:

```
surfc(x, y, z)
meshc(x, y, z)
```

在曲面下面创建等高面。x,y,z 为确定面的 x,y,z 坐标。这两个函数的介绍参见表 7.5。

可创建不带有曲面的各种等高线图,并可选择是否对其进行标注。函数:

```
contour(x,y,z,v)
```

创建二维等高线图。x,y,z 为定义面的点坐标。如果 v 为标量,则表示显示的等高线数。而如果 v 为向量,则表示各曲面等高线值。v 是可选项。可使用以下函数对标注等高线:

```
[C,h] = contour(x,y,z,v)
clabel(C,h,v)
```

表7.5 meshc 和 surfc 函数的说明

绘图函数	程序	图形
meshc	[x,y,z] = SurfExample; meshc(x,y,z) grid off	
surfc	[x,y,z] = SurfExample; surfc(x,y,z) grid off	

表7.6给出了这两个函数的介绍。

此外,尚有两个函数可实现等高线的绘制。第一个函数为:

```
contour3(x,y,z,v)
```

用于绘制三维等高线。x, y, z 是定义面的点的坐标。如果 v 为标量,则表示显示的等高线数。而如果 v 为向量,则表示各曲面等高线值。v 是可选项。标注等高线可使用以下函数对:

```
[C,h] = contour3(x,y,z,v)
clabel(C, h, v)
```

第二个函数为:

```
contourf(x,y,z,v)
```

可用不同颜色填充二维等高线之间的区域。填充颜色可用函数

```
colorbar(s)
```

定义。colorbar 生成一个彩色条状图,并在图上标出相关数值。字符串 s 取值为"horiz"或"vert",用以表明条状图的方向。默认值为"vert"。表7.7给出了这两个函数的介绍。

柱面图

可对二维曲线进行处理以生成曲面。函数

```
[x,y,z] = cylinder(r, n)
```

依据向量 r 定义的轮廓线返回柱面的 x,y,z 坐标。函数 cylinder 对其圆周进行 n 等分,并将向量 r 中的各元素值作为各等分点的半径。如果省略 n,则 MATLAB 默认值为 20。

表 7.6 contour 和 clabel 函数的说明

绘图函数	程序	图形
contour	[x,y,z] = SurfExample; contour(x,y,z)	
contour	[x,y,z] = SurfExample; contour(x,y,z,4)	
contour clabel	[x,y,z] = SurfExample; [C,h] = contour(x,y,z); clabel(C,h)	
contour clabel	[x,y,z] = SurfExample; v = [10,30:30:120]; [C,h] = contour(x,y,z,v); clabel(C,h,v)	

表7.7 coutour3,contourf 和 colorbar 函数说明

绘图函数	程序	图形
contour3 clabel	[x,y,z] = **SurfExample**; [C,h] = contour3(x,y,z); clabel(C,h)	
contourf	[x,y,z] = **SurfExample**; contourf(x,y,z) colorbar	

为说明函数 cylinder 的用法,考虑以下曲线:

$$r = 1.1 + \sin(z) \quad 0 \leqslant z \leqslant 2\pi$$

将其沿 z 轴旋转 360°,并在 z 轴方向进行 26 等分,在圆周方向进行 16 等分。则绘制柱面图形的程序为:

```
zz = linspace(0,2 * pi,26);
[x,y,z] = cylinder(1.1 + sin(zz),16);
surf(x,y,z)
axis off
```

结果如图 7.5 所示。

视角

图 7.5 中,视角为默认值。然而,有时基于以下几点原因,希望改变其默认视角:(1)图形未显示出期望的曲面特征;(2)希望用 subplot 显示曲面的几种不同视图;(3)决定最后结果之前希望对曲面进行多视角观察。可利用函数 view 确定图形的方位角和仰角:

```
[a,e] = view
```

其中,a 为方位角,e 为仰角。为确定图形视角,可在图形窗口中点击旋转图标直至达到满意

的视觉效果。在命令窗口中输入上述语句后将显示图形的方位角和仰角值。将其代入表达式：

```
view (an,en)
```

则在下次程序运行时可创建理想的视图。表达式中的 *an* 和 *en* 是从命令窗口获得的 *a* 和 *e* 值。

在图 7.5 所示的图中应用上述过程，可发现当 $a = -88.5°$，$e = -48°$时可获得满意的视图效果。修改后的程序及执行结果参见表 7.8。



图 7.5　函数 cylinder 的应用

表 7.8　view 和 shading 函数的说明

绘图函数	程序	图　形

view
shading‘faceted’
	zz = linspace(0,2 * pi,26);
	r = 1 + sin(zz);
	[x,y,z] = cylinder(r,16);
	surf(x,y,z)
	view(- 88.5, - 48)
	shading‘faceted’
	axis off	

shading‘flat’	zz = linspace(0,2 * pi,26);
	r = 1 + sin(zz);
	[x,y,z] = cylinder(r,16);
	surf(x,y,z)
	view(- 88.5, - 48)
	shading‘flat’
	axis off	

(续表)

绘图函数	程序	图 形

shading 'interp'	zz = linspace(0,2 * pi,26);
	r = 1 + sin (zz);
	[x,y,z] = cylinder(r,16);
	surf(x,y,z)
	view(- 88.5, - 48)
	shading 'interp'
	axis off	

阴影

函数 surf 创建的曲面具有默认的阴影特性值“faceted”。改变图形阴影特性的函数为:

```
shading s
```

其中,字符串 s 值为“faceted”,“flat“或“interp”。各值的效果图如表 7.8 所示。

下面的示例说明三维绘图函数的应用。

例 7.3 平面的生成

当空间内的 3 个点:$\mathbf{P}_o(x_o,y_o,z_o)$,$\mathbf{P}_1(x_1,y_1,z_1)$和 $\mathbf{P}_2(x_2,y_2,z_2)$确定时,则包含此三点的平面参数方程为:

$$\mathbf{P} = \mathbf{P}_o + s\mathbf{v} + t\mathbf{w}$$

其中,

$$\mathbf{P} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

$$\mathbf{P}_o = x_o\mathbf{i} + y_o\mathbf{j} + z_o\mathbf{k}$$

$$\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k} = (x_1 - x_o)\mathbf{i} + (y_1 - y_o)\mathbf{j} + (z_1 - z_o)\mathbf{k}$$

$$\mathbf{w} = w_1\mathbf{i} + w_2\mathbf{j} + w_3\mathbf{k} = (x_2 - x_o)\mathbf{i} + (y_2 - y_o)\mathbf{j} + (z_2 - z_o)\mathbf{k}$$

$0 \leqslant s \leqslant 1, 0 \leqslant t \leqslant 1$。因此,

$$x = x_o + sv_1 + tw_1 = x_o + s(x_1 - x_o) + t(x_2 - x_o)$$

$$y = y_o + sv_2 + tw_2 = y_o + s(y_1 - y_o) + t(y_2 - y_o)$$

$$z = z_o + sv_3 + tw_3 = z_o + s(z_1 - z_o) + t(z_2 - z_o)$$

假设使用 5×5 的网格能充分显示该平面,则可创建 **PlanarSurface** 函数以生成并显示此平面:

```
function[xx,yy,zz,L] = PlanarSurface(P0,P1,P2)
v = P1 - P0;
w = P2 - P0;
S = 0:0.2:1;
L = length(S);
[s,t] = meshgrid(S,S);
xx = P0(1) + s * v(1) + t * w(1);
yy = P0(2) + s * v(2) + t * w(2);
zz = P0(3) + s * v(3) + t * w(3);
```

其中,P_0,P_1 和 P_2 是表示平面上 3 点坐标的 3 元素向量。因此,如果在 MATLAB 命令窗口输入以下语句:

```
[x,y,z,L] = PlanarSurface([0 0 0],[2 6 3],[7 1 5]);
surf(x,y,z)
```

则可得到图 7.6。

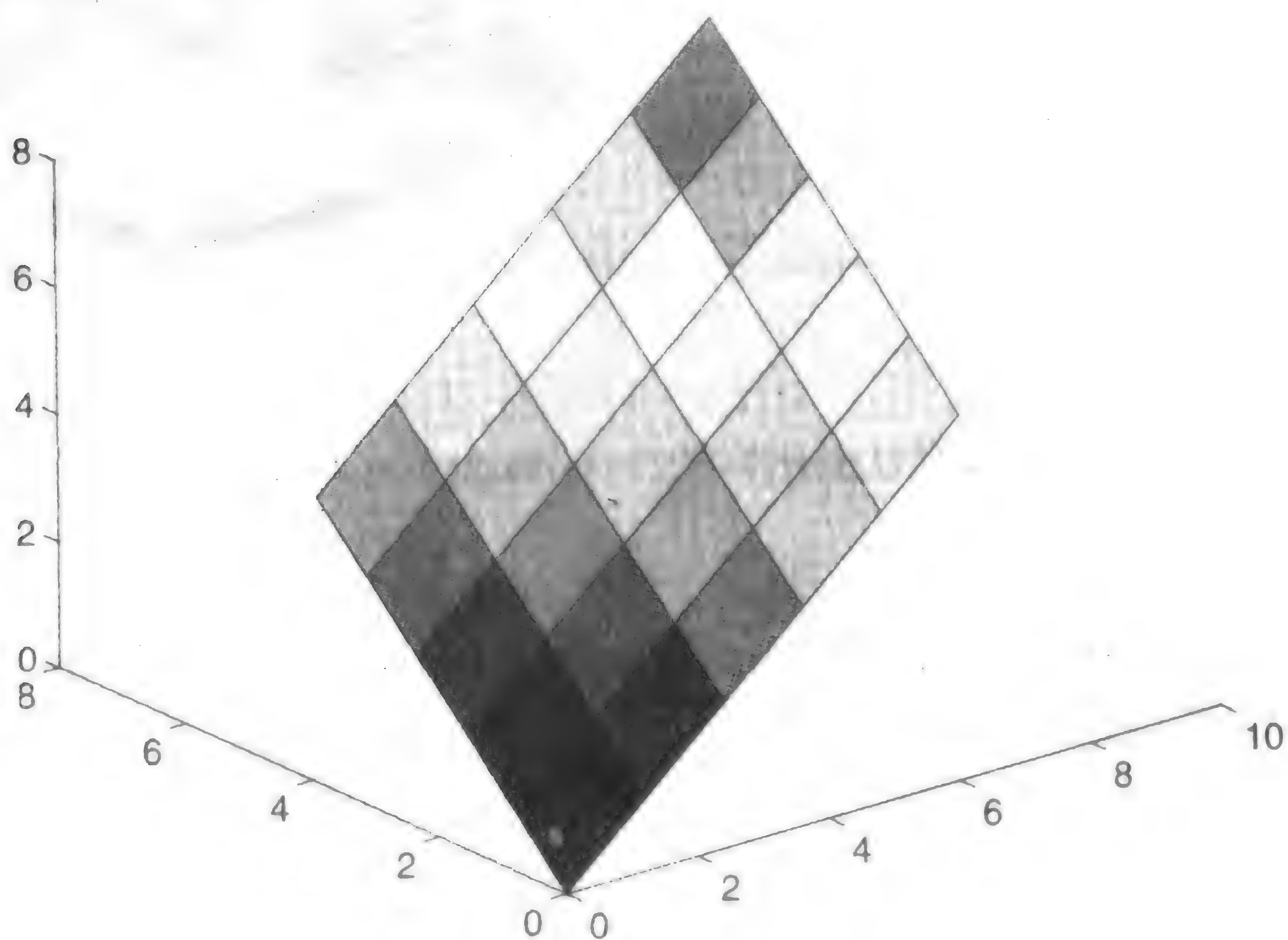


图 7.6 平面的生成

为将曲面投影到三个相互垂直的坐标参考面上,应采用适当的点乘运算。因此,为将平面投影到 xy 平面上,则:

$$\mathbf{P}\cdot(\mathbf{i}+\mathbf{j}+0\mathbf{k})$$

与之类似,如将平面投影到 yz 平面上,则:

$$\mathbf{P}\cdot(0\mathbf{i}+\mathbf{j}+\mathbf{k})$$

如将平面投影到 xz 平面上,则:

$$\mathbf{P}\cdot(\mathbf{i}+0\mathbf{j}+\mathbf{k})$$

为此,创建新函数 **PlanarSurfaceProj** 以包含这些投影:

```
function PlanarSurfaceProj(P0,P1,P2)
[xx,yy,zz,L] = PlanarSurface(P0,P1,P2);
hold on
a = axis;
c(1:L,1:L,1:3) = zeros(L,L,3);
c(:,:,1) = 1;
c(:,:,2) = 1;
c(:,:,3) = 0;
surf(xx,yy,a(5) * ones(L,L),c)
surf(xx,a(4) * ones(L,L),zz,c)
surf(a(2) * ones(L,L),yy,zz,c)
```

数组 c 的定义使投影面显示为黄色。数组 c 的前两个索引必须具有相同阶次，如 xx，yy 和 zz。最后的索引准确代表了 3 个元素，其中每个元素的取值范围都从 0 到 1。最后三个元素定义了前两个索引的每种组合下每个色块的颜色。执行程序：

```
[x,y,z,L] = PlanarSurface([0 0 0],[2 6 3],[7 1 5]);
surf(x,y,z)
PlanarSurfaceProj([0 0 0],[2 6 3],[7 1 5])
```

结果得到图 7.7。图中似乎给人一种错觉，即投影面并不在设计面上，但是用 *Rotate 3D* 可证实它们确实在相应的坐标平面上。

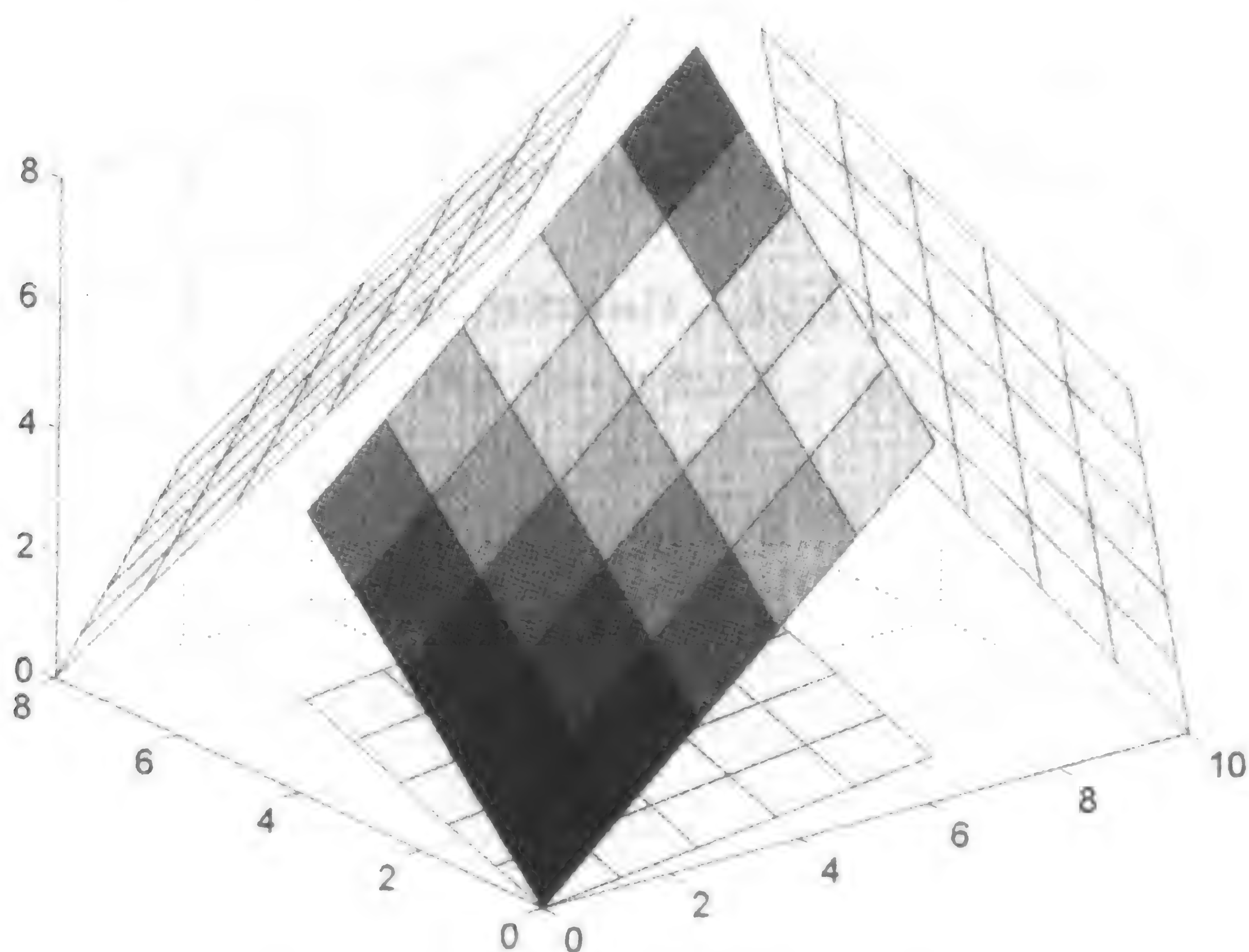


图 7.7　将一个平面投影到参考坐标面上

例 7.4　盒子的生成

用例 7.3 的结果生成长方体(盒子)，其尺寸为 $L_x \times L_y \times L_z$。参见图 7.1，由图中可看出盒子由 6 个面组成，且每个面由以下 3 组点定义。

垂直于 yz 平面的面：

$$\mathbf{P}_o(x_o, y_o, z_o), \mathbf{P}_1(x_o, y_o, z_o + L_z), \mathbf{P}_2(x_o, y_o + L_y, z_o)$$

$$\mathbf{P}_o(x_o + L_x, y_o, z_o), \mathbf{P}_1(x_o + L_y, y_o, z_o + L_z), \mathbf{P}_2(x_o + L_x, y_o + L_y, z_o)$$

垂直于 xz 平面的面：

$$\mathbf{P}_o(x_o, y_o, z_o), \mathbf{P}_1(x_o, y_o, z_o + L_z), \mathbf{P}_2(x_o + L_y, y_o, z_o)$$

$$\mathbf{P}_o(x_o, y_o + L_y, z_o), \mathbf{P}_1(x_o, y_o + L_y, z_o + L_z), \mathbf{P}_2(x_o + L_x, y_o + L_y, z_o)$$

垂直于 xy 平面的面：

$$\mathbf{P}_o(x_o, y_o, z_o), \mathbf{P}_1(x_o + L_x, y_o, z_o), \mathbf{P}_2(x_o, y_o + L_y, z_o)$$

$$\mathbf{P}_o(x_o, y_o, z_o + L_z), \mathbf{P}_1(x_o + L_x, y_o, z_o + L_z), \mathbf{P}_2(x_o, y_o + L_y, z_o + L_z)$$

在函数 **BoxSurface** 中应用上述关系：

```
function BoxSurface(P0,L)
[x,y,z] = PlanarSurface(P0,P0+[0 0 L(3)],P0+[0 L(2) 0]);
surf(x,y,z)
hold on
[x,y,z] = PlanarSurface(P0+[L(1) 0 0],P0+[L(1) 0 L(3)],P0+[L(1) L(2) 0]);
surf(x,y,z)
[x,y,z] = PlanarSurface(P0,P0+[0 0 L(3)],P0+[L(1) 0 0]);
surf(x,y,z)
[x,y,z] = PlanarSurface(P0+[0 L(2) 0],P0+[0 L(2) L(3)],P0+[L(1) L(2) 0]);
surf(x,y,z)
[x,y,z] = PlanarSurface(P0,P0+[L(1) 0 0],P0+[0 L(2) 0]);
surf(x,y,z)
[x,y,z] = PlanarSurface(P0+[0 0 L(3)],P0+[L(1) 0 L(3)],P0+[0 L(2) L(3)]);
surf(x,y,z)
```

其中，$\mathbf{P}_o=[x_o,y_o,z_o]$，$L=[L_x,L_y,L_z]$，**PlanarSurface** 在例 7.3 中给出。

以下程序生成三个盒子，盒子的尺寸及其中一个角的位置如下所示：

盒子 #1:

尺寸:3×5×7

位置:(1,1,1)

盒子 #2:

尺寸:4×5×1

位置:(3, 4, 5)

盒子 #3:

尺寸:1×1×1

位置:(4.5,5.5,6)

创建并显示这些盒子的程序如下：

```
BoxSurface([1,1,1],[3,5,7])
BoxSurface([3,4,5],[4,5,1])
BoxSurface([4.5,5.5,6],[1,1,1])
view(29.5,44)
```

结果如图 7.8 所示。

例 7.5　三维物体的旋转和转换：欧拉角

从一点 $p(x,y,z)$ 到另一点 $P(X,Y,Z)$ 的旋转和转换为[①]：

$$X=L_x+a_{11}x+a_{12}y+a_{13}z$$

$$Y=L_y+a_{21}x+a_{22}y+a_{23}z$$

$$Z=L_z+a_{21}x+a_{22}y+a_{23}z$$

其中，L_x，L_y 和 L_z 分别为转换的 (x,y,z) 分量，a_{ij} 为下列矩阵的元素，其中，$i,j=1,2,3$：

① 参见 W. Gellert, H. Kustner, M. Hellwich 和 H. Kastner, *The VNR Concise Encyclopedia of Mathematics*, Van Nostrand Reinhold, New York, NY, 1975 年，第 534～535 页。

$$a=\begin{bmatrix}\cos\Psi\cos\chi & -\cos\Psi\sin\chi & \sin\Psi\\ \cos\phi\sin\chi+\sin\phi\sin\Psi\cos\chi & \cos\phi\cos\chi-\sin\phi\sin\Psi\sin\chi & -\sin\phi\cos\Psi\\ \sin\phi\sin\chi-\cos\phi\sin\Psi\cos\chi & \sin\phi\cos\chi+\cos\phi\sin\Psi\sin\chi & \cos\phi\cos\Psi\end{bmatrix}$$

ϕ, Ψ 和 χ 是坐标系关于原点的顺序旋转角(欧拉角):ϕ 为 x 轴的旋转角,Ψ 为 y 轴的旋转角,χ 为 z 轴的旋转角。一般而言,(x,y,z)可以是标量、等长向量或等阶次矩阵。

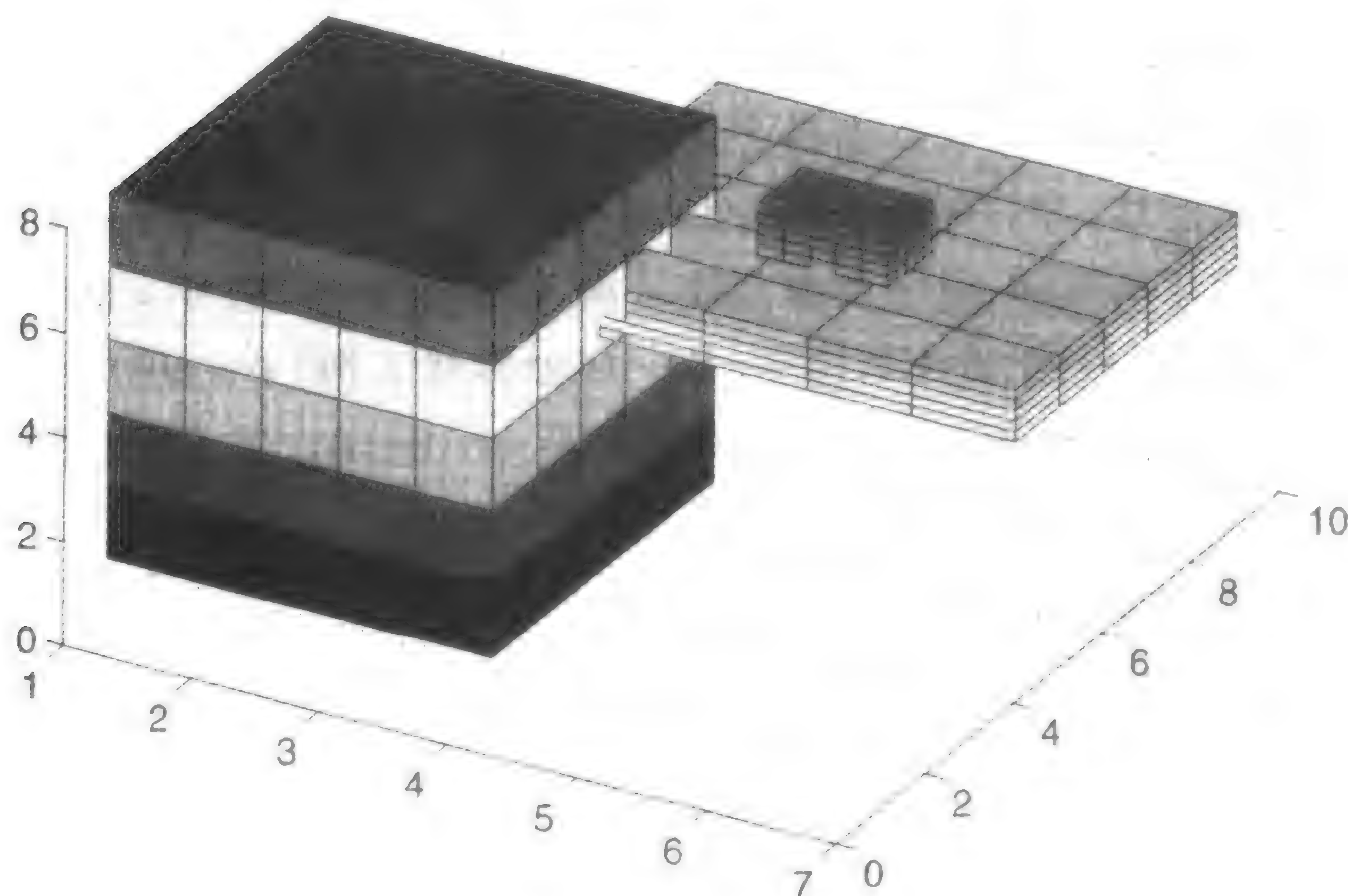


图 7.8 三个方盒

在应用这些关系之前,先创建以下函数 **EulerAngles**:

```
function[Xrt,Yrt,Zrt] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,x,y,z)
a = [cos(psi) * cos(chi), - cos(psi) * sin(chi),sin(psi);
    cos(phi) * sin(chi) + sin(phi) * sin(psi) * cos(chi),cos(phi) * cos(chi) -
    sin(phi) * sin(psi) * sin(chi),-sin(phi) * cos(psi);
    sin(phi) * sin(chi) - cos(phi) * sin(psi) * cos(chi),
    sin(phi) * cos(chi) + cos(phi) * sin(psi) * sin(chi),cos(phi) * cos(psi)];
Xrt = a(1,1) * x + a(1,2) * y + a(1,3) * z + Lx;
Yrt = a(2,1) * x + a(2,2) * y + a(2,3) * z + Ly;
Zrt = a(3,1) * x + a(3,2) * y + a(3,3) * z + Lz;
```

现在,以对圆环面的变换为例,说明这些变换方程的用法,其坐标为[①]:

$$x=r\cos\theta$$
$$y=r\sin\theta$$
$$z=\pm\sqrt{a^2-(\sqrt{x^2+y^2}-b)^2}$$

$b-a\leqslant r\leqslant b+a$,$0\leqslant\theta\leqslant 2\pi$,且 $b>a$。

首先,创建函数 **Torus** 获得圆环面坐标:

① 参见 D. von Seggern,出处同上。

```
function[X,Y,Z] = Torus(a,b)
r = linspace(b - a,b + a,10);
th = linspace(0,2 * pi,22);
x = r' * cos(th);
y = r' * sin(th);
z = real(sqrt(a^2 - (sqrt(x.^2 + y.^2) - b).^2));
X = [x x];
Y = [y y];
Z = [z -z];
```

其中,real 用于消除由数值方法产生的虚部。

可获得四个圆环面图形。第一个图形是没有旋转的圆环面图形。第二个图形中包含两部分,一部分为原始圆环,另一部分为绕 x 轴旋转 60°的圆环($\phi = 60°$)。第三个图形包含了两部分,一部分为原始圆环,另一部分为绕 y 轴旋转 60°的圆环($\Psi = 60°$)。最后一个图形同样包含了两部分,一部分为原始圆环,另一部分为绕 x 轴($\phi = 60°$)和 y 轴($\Psi = 60°$)都旋转 60°的圆环。假设 $a = 0.2$, $b = 0.8$,利用函数 colormap 生成黑色网格线。程序为:

```
a = 0.2; b = 0.8;
[X,Y,Z] = Torus(a,b);
Lx = 0; Ly = 0; Lz = 0;
for k = 1:4
  subplot(2,2,k)
  switch k
    case 1
      mesh(X,Y,Z)
      v = axis
      axis([v(1) v(2) v(3) v(4) -1 1])
      text(0.5, -0.5,1,'Torus')
    case 2
      psi = 0; chi = 0; phi = pi/3;
      [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
      mesh(X,Y,Z)
      hold on
      mesh(Xr,Yr,Zr)
      text(0.5, -0.5,1,'\phi = 60 \circ')
    case 3
      psi = pi/3; chi = 0; phi = 0;
      [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
      mesh(X,Y,Z)
      hold on
      mesh(Xr,Yr,Zr)
      text(0.5, -0.5,1,'\psi = 60 \circ')
    case 4
      psi = pi/3; chi = 0; phi = pi/3;
      [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
      mesh(X,Y,Z)
      hold on
      mesh(Xr,Yr,Zr)
      text(0.5, -0.5,1.35,'\psi = 60 \circ')
      text(0.55, -0.5,1,'\phi = 60 \circ')
  end
```

```
    colormap([0 0 0])
    axis equal
    axis off
    grid off
end
```

程序结果如图 7.9 所示。

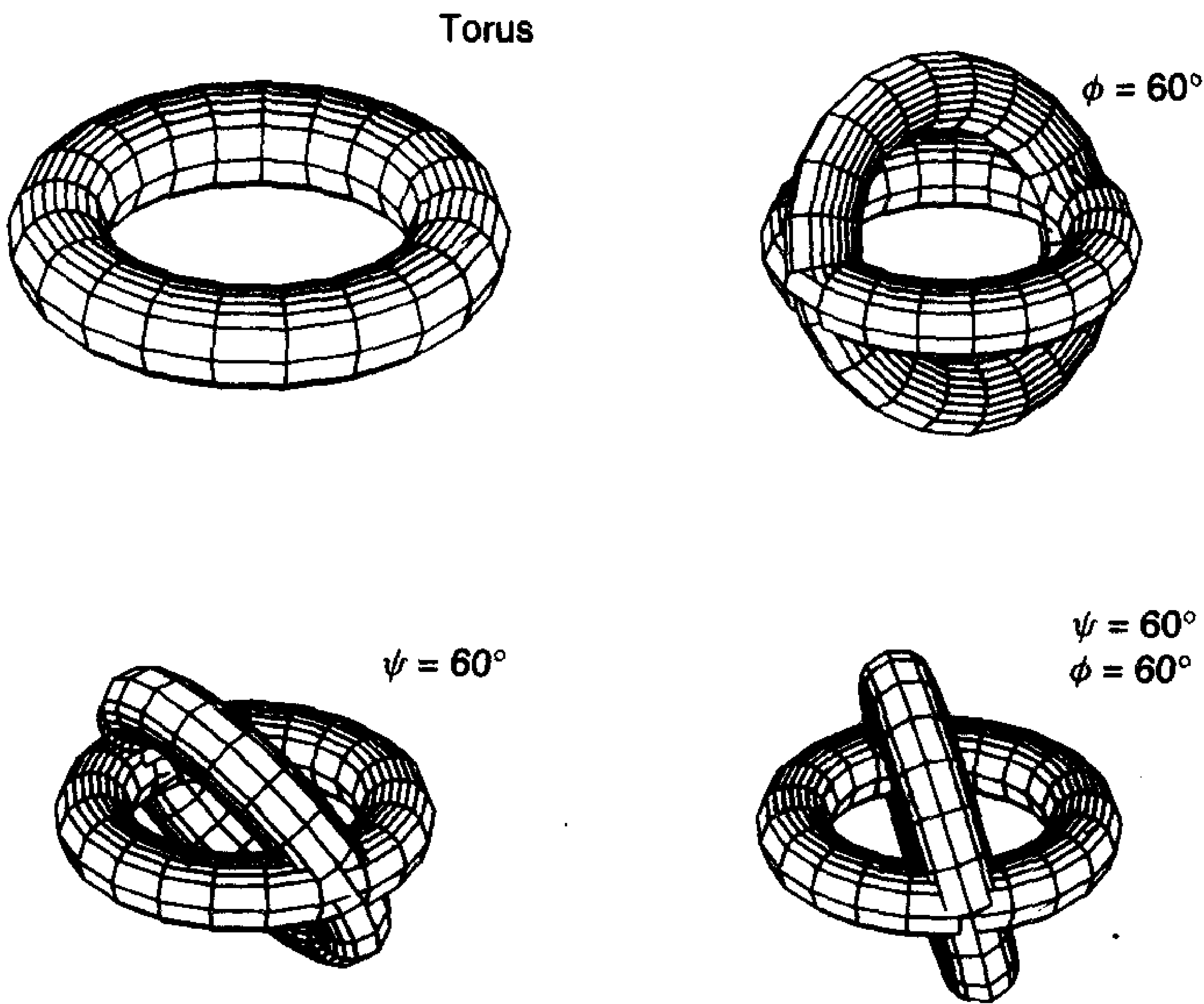


图 7.9 圆环的旋转

7.3 应用函数一览

表 7.9 给出了本章应用到的绘图函数介绍。

表 7.9 第 7 章应用的 MATLAB 函数

MATLAB 函数	说明
axis on/off	打开和关闭轴
box on/off	在轴的边框放置盒子(必须选定 axis on)
clabel	标注等高线高度
colorbar	用设定颜色显示颜色条状图
colormap	用三元向量设定色彩图
contour	创建二维等高线
contourf	用彩色填充二维等高线区域
contour3	创建三维等高线图
cylinder	生成圆柱特定剖面坐标
grid on/off	打开或关闭图形网格线
mesh	用白色方块绘制曲面,线段颜色基于 z 轴方向值
meshc	实现 mesh 函数功能并在曲面下包含等高线

(续表)

MATLAB 函数	说明
meshz	在曲面顶点处绘制垂直平面
plot3	绘制三维图形
shading	设置 surf 函数创建曲面的阴影特性值
surf	用彩色方块绘制曲面,颜色值基于 z 轴方向值
surfc	实现 surf 函数功能并在曲面下包含等高线
view	改变视图视角
waterfall	实现 mesh 函数创建曲面的波纹效果
zlabel	标注 z 轴

练习

7.1 使用 axis equal 语句画出以下三维曲线[①]。

(a) 球面螺旋线($c=5.0, 0\leqslant t\leqslant 10\pi$):

$$x=\sin(t/2c)\ \cos(t)$$
$$y=\sin(t/2c)\ \sin(t)$$
$$z=\cos(t/2c)$$

(b) 柱面正弦波($a=10.0, b=1.0, c=0.3, 0\leqslant t\leqslant 2\pi$):

$$x=b\ \cos(t)$$
$$y=b\ \sin(t)$$
$$z=c\ \cos(at)$$

(c) 球面正弦波($a=10.0, b=1.0, c=0.3, 0\leqslant t\leqslant 2\pi$):

$$x=\cos(t)\sqrt{b^2-c^2\cos^2(at)}$$
$$y=\sin(t)\sqrt{b^2-c^2\cos^2(at)}$$
$$z=c\cos(at)$$

(d) 环面螺旋线($a=0.2, b=0.8, c=20.0, 0\leqslant t\leqslant 2\pi$):

$$x=[b+a\sin(ct)]\cos(t)$$
$$y=[b+a\sin(ct)]\sin(t)$$
$$z=a\cos(ct)$$

7.2 画出以下实体的曲面[②]。对(a)到(f)中的坐标,适当时可采用坐标变换 $x=r\cos(\theta)$, $y=r\sin(\theta)$或 $x=a\cos(\theta)$, $y=b\sin(\theta)$得到的向量形式,其中$0\leqslant\theta\leqslant 2\pi$。令 r(或 a 与/或 b)为 10,步长 θ 为 22。练习(f)要求使用 meshgrid 函数,所有图形要求使用 axis equal 语句。

---

① 参见 D. von Seggern,出处同上。

② 参见 D. von Seggern,出处同上。

(a) 球体($r=1$):

$$z=\sqrt{r^2-x^2-y^2}$$

(b) 椭圆体($a=1.0, b=1.5, c=2.0$):

$$z=c\sqrt{1-x^2/a^2-y^2/b^2}$$

(c) 扁球体(椭圆体 $a=b>c: a=b=1.0, c=0.5$)

(d) 扁长球体(椭圆体 $a=b<c: a=b=1.0, c=1.2$)

(e) 圆锥体($0\leqslant r\leqslant 2$)[必须使用两次 surf 函数]:

$$z=\pm\sqrt{x^2+y^2}$$

(f) 丰饶角($a=0.3, b=0.5, 0\leqslant u\leqslant 2\pi, -3\leqslant v\leqslant 3$)[用 *Rotate 3D* 分析曲面]:

$$x=e^{bv}\cos(v)+e^{av}\cos(u)\cos(v)$$
$$y=e^{bv}\sin(v)+e^{av}\cos(u)\sin(v)$$
$$z=e^{av}\sin(u)$$

7.3 用 surf 函数画出下列曲面[1],并用 *Rotate 3D* 图标进行分析。

(a) ($a=b=1, c=0.5, -3\leqslant x\leqslant 3, -3\leqslant y\leqslant 3$):

$$z=c((x/a)^4\pm(y/b)^4)$$

(b) ($a=3, c=0.25, -1\leqslant x\leqslant 1, -1\leqslant y\leqslant 1$):

$$z=c\sin(2\pi a\sqrt{x^2+y^2})$$

(c) ($a=3, c=0.25, -1\leqslant x\leqslant 1, -1\leqslant y\leqslant 1$):

$$z=c\sin(2\pi axy)$$

(d) ($c=0.2, -1\leqslant x\leqslant 1, -1\leqslant y\leqslant 1, x\neq 0, y\neq 0$):

$$z=c\ln(|xy|)$$

(e) 悬索曲面($1\leqslant u\leqslant 5, 0\leqslant v\leqslant 2\pi$):

$$x=u\cos(v)$$
$$y=u\sin(v)$$
$$z=\cosh^{-1}(u)$$

(f) 正螺旋面($c=1/2\pi, -0.5\leqslant u\leqslant 0.5, -2\pi\leqslant v\leqslant 2\pi$):

$$x=u\cos(v)$$
$$y=u\sin(v)$$
$$z=cv$$

(g) 双曲螺旋面[2] [$\tau=7, -\pi\leqslant u\leqslant\pi$(在该范围内取 150 个值),$0\leqslant v\leqslant 0.5$(在该范围内取 6 个值)]:

$$x=\frac{\sinh(v)\cos(\tau u)}{1+\cosh(u)\cosh(v)}$$

$$y=\frac{\sinh(v)\sin(\tau u)}{1+\cosh(u)\cosh(v)}$$

---

① 参见 D. von Seggern,出处同上。

② 参见 E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*,第二版,Chapman & Hall/CRC, Boca Raton, FL, 2003 年,第 1421 页。

$$z=\frac{\sinh(u)\cosh(v)}{1+\cosh(u)\cosh(v)}$$

(h) 星形椭圆体[①] ($a=b=c=1,-\pi/2\leqslant u\leqslant\pi/2,-\pi\leqslant u\leqslant\pi$):

$$x=(a\ \cos(u)\ \cos(v))^3$$
$$y=(b\ \sin(u)\ \cos(v))^3$$
$$x=(c\ \sin(v))^3$$

(i) 麦比乌斯带[②] ($-0.4\leqslant s\leqslant 0.4,0\leqslant v\leqslant 2\pi$):

$$x=s\ \cos(t/2)\ \cos(t)$$
$$y=s\ \cos(t/2)\ \sin(t)$$
$$z=s\ \sin(t/2)$$

(j) 正弦曲面[③] ($0\leqslant u\leqslant 2\pi,0\leqslant v\leqslant 2\pi$):

$$x=a\ \sin(u)$$
$$y=a\ \sin(v)$$
$$z=a\ \sin(u+v)$$

(k) 惠特尼伞[④] ($-1\leqslant u,\ v\leqslant 1$):

$$x=uv$$
$$y=u$$
$$z=v^2$$

(l) 螺旋弹簧[⑤] ($r_1=r_2=0.25,T=2,n=4,0\leqslant u,\ v\leqslant 2\pi$),应用函数 view([-64 0]):

$$x=[1-r_1\ \cos(v)]\cos(nu)$$
$$y=[1-r_1\ \cos(v)]\sin(u)$$
$$z=r_2[\sin(v)+Tu/\pi]$$

(m) 月牙形[⑥] ($0\leqslant u,\ v\leqslant 1$):

$$x=(2+\sin(2\pi u)\ \sin(2\pi v))\ \sin(3\pi v)$$
$$y=(2+\sin(2\pi u)\ \sin(2\pi v))\ \cos(3\pi v)$$
$$z=\cos(2\pi u)\ \sin(2\pi v)+4v-2$$

(n) 图8圆环[⑦] ($c=1,-\pi\leqslant u,\ v\leqslant\pi$):

$$x=\cos(u)(c+\sin(v)\cos(u)-\sin(2v)\sin(u)/2);$$
$$y=\sin(u)(c+\sin(v)\cos(u)-\sin(2v)\sin(u)/2);$$
$$z=\sin(u)\sin(v)+\cos(u)\sin(2v)/2;$$

① 参见 E. W. Weisstein,出处同上,第136页。
② 参见 E. W. Weisstein,出处同上,第1928页。
③ 参见 E. W. Weisstein,出处同上,第2708页。
④ 参见 E. W. Weisstein,出处同上,第3200页。
⑤ 参见 http://astronomy.swin.edu.au/pbourke/geometry/。
⑥ 参见 http://astronomy.swin.edu.au/pbourke/geometry/。
⑦ 参见 http://astronomy.swin.edu.au/pbourke/geometry/。

7.4 一个边缘固定的实心圆盘，$r=b$，其模型外形为[1]：

$$w_{mn}(r,\theta)=[C_{mn}J_m(\Omega_{mn}r/b)+I_m(\Omega_{mn}r/b)]\cos(m\theta)$$

其中，$m=0,1,2,\cdots$，$J_m(x)$为一类 m 阶贝塞尔函数，$I_m(x)$为一类 m 阶贝塞尔函数的修正函数：

$$C_{mn}=-\frac{I_m(\Omega_{mn})}{J_m(\Omega_{mn})}$$

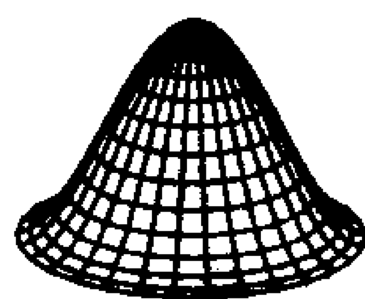
Ω_{mn}为下式的解：

$$J_m(\Omega_{mn})I_{m+1}(\Omega_{mn})+I_m(\Omega_{mn})J_{m+1}(\Omega_{mn})=0$$

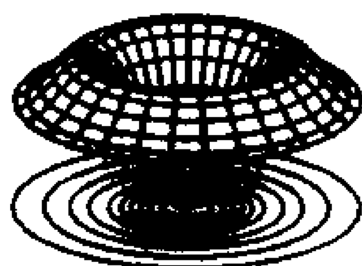
并且在练习 5.9 中已求出。

利用练习 5.9 中，当 $m=0,1,2$ 时所得到的三个最低自然频率系数，使用 surfc 和 subplot 函数在同一图形中画出相应的 9 种模型外形，并在每个图形的上方标注 m，n 值及相应的频率系数。不要画出 x 轴和 y 轴。第一行图形为 $m=0$ 时的情形，依次类推。使用两次 max 函数（因为位移域是矩阵）对每个模型外形进行标准化处理，使其最大幅值为 1。建议半径分数（r/b）为 15，角分数为 30。结果如图 7.10 所示。运用 meshc 函数和 colormap 函数提高复制图形的清晰度。

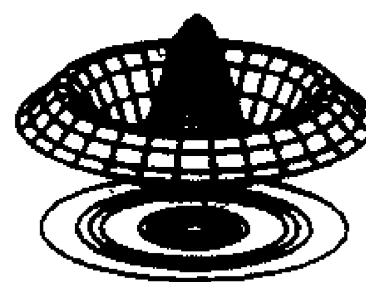
3.1962 m=0 n=1



6.3064 m=0 n=2



9.4395 m=0 n=3



4.6109 m=1 n=1



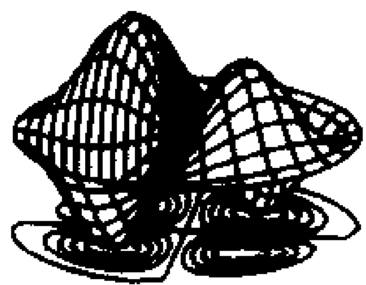
7.7993 m=1 n=2



10.9581 m=1 n=3



5.9057 m=2 n=1



9.1969 m=2 n=2



12.4022 m=2 n=3



图 7.10 圆盘的变形

7.5 一块厚平板，x 方向的厚度为 $2L$，y，z 方向均很长。$t=0$ 时刻其为恒温 T_i，如果突然将厚平板置于温度为 T_∞ 的对流环境中，则其温度分布随时间与位置变化的函数关系为[2]：

---

① 参见 E. B. Magrab, *Vibration of Elastic Structural Members*, Sijthoff & Noordhoff, The Netherlands, 1997 年，第 252 页。

② 参见 D. R. Pitts and L. E. Sissom, *Theory and Practice of Heat Transfer*, Schaum's Outline Series, McGraw-Hill, New York, NY, 1997 年，第 79 页。

$$\frac{\theta}{\theta_i} = 2\sum_{n=1}^{\infty} \frac{\sin\delta_n\cos(\delta_n\eta)}{\delta_n + \sin\delta_n\cos(\delta_n)}\exp(-\delta_n^2\tau)$$

其中,$\theta = \theta(\eta,\tau) = T - T_\infty$,$T = T(\eta,\tau)$为厚平板温度,$\theta_i = T_i - T_\infty$,$\eta = x/L$,$\tau = \alpha^2 t/L^2$为无量纲时间(有时称为傅里叶系数),$\alpha$ 为热扩散系数,δ_n 为下式的解:

$$\cot\delta_n = \frac{\delta_n}{B_i}$$

其中,$B_i = \bar{h}L/k$ 为毕奥特(Biot)数,$\bar{h}$ 为全部曲面上对流的平均传热系数,k 为厚平板的热导。

在 $B_i = 0.7$ 时求出 δ_n 的 20 个最低值,并用该值在 $0 \leqslant \eta \leqslant 1$ 及 $0 \leqslant \tau \leqslant 2$ 时画出曲面 $\theta(\eta,\tau)/\theta_i$。然后,用 *Rotate 3D* 函数交替得到一个可接受的曲面视图。标注轴并给出图形标题,加上垂直线以进一步加强曲面的可视化,如图 7.4 所示。

7.6 画出下式所表示的外缘固定的方形薄膜形状及等高线图,在 $0 \leqslant x \leqslant 1$ 及 $0 \leqslant y \leqslant 1$ 时,图中的每个方向使用 25 个网格点:

$$w_{23}(x,y) = \sin(2\pi x)\sin(3\pi y)$$

7.7 一块长为 l 的平板,流过平板的湍流的平均努塞尔特(Nusselt)数为[①]:

$$Nu = \frac{0.037\mathrm{Re}^{0.8}\mathrm{Pr}}{1 + 2.443\mathrm{Re}^{-0.1}(\mathrm{Pr}^{2/3} - 1)} \quad 5\times10^5 \leqslant \mathrm{Re} \leqslant 10^7 \quad 0.6 \leqslant \mathrm{Pr} \leqslant 2000$$

其中,Re 为雷诺(Reynolds)数,Pr 为普朗特(Prandtl)数。画出 lg(Nu)在所示范围内随 lg(Re)和 lg(Pr)变化的函数曲面。如图 7.4 所示,用垂线将图中平板的边界和曲面各角进行连接。

7.8 如图 7.11 所示,钢筋混凝土梁的中轴位置由参数 k 决定,k 值如下[②]:

$$k = -\rho n + \sqrt{(\rho n)^2 + 2\rho n}$$

其中,$\rho = A_s/bd$,$n = E_s/E_c$ 为钢筋和混凝土的弹性模量之比。画出 k 随 n 与 ρ 变化的曲面图。其中,在 $6 \leqslant n \leqslant 12$ 内取 n 的 10 个值,在 $0.001 \leqslant \rho \leqslant 0.009$ 内取 ρ 的 9 个值,并在 $0.01 \leqslant \rho \leqslant 0.1$ 内取 ρ 的 10 个值。

7.9 受扭力 T 作用的矩形梁,其最大无量纲主剪切应力 τ' 由下式得到[③]:

$$\tau'^2 = \tau'^2_{xz} + \tau'^2_{yz}$$

其中,

$$\tau'_{xz} = \frac{\tau_{xz}J}{Ta} = -\frac{16}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)^2}\frac{\sinh(k_n\xi)}{\cosh(k_n b/a)}\cos(k_n\eta)$$

$$\tau'_{yz} = \frac{\tau_{yz}J}{Ta} = 2\eta - \frac{16}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)^2}\frac{\cosh(k_n\xi)}{\cosh(k_n b/a)}\sin(k_n\eta)$$

---

① 参见 W. Beitz 和 K. H. Kuttner 编辑, *Handbook of Mechanical Engineering*, Springer-Verlag, New York, NY, 1994 年, 第 C31 页。

② 参见 L. Spiegal 和 G. F. Limbrunner, *Reinforced Concrete Design*, 第三版, Prentice Hall, Upper Saddle River, NJ, 1992 年, 第 196 页。

③ 参见 C. T. Wang, *Applied Elasticity*, McGraw-Hill, New York, NY, 1953 年, 第 89 页。

J 为转矩常数，$\eta = x/a(-1 \leqslant \eta \leqslant 1)$，$\xi = y/b(-1 \leqslant \xi \leqslant 1)$，$k_n = (2n+1)\pi/2$。在 $b/a = 1$ 时，生成 $\tau'^2(\eta,\xi)$ 所表示的曲面，为方形横截面。在另一幅图中画出 $\tau'^2(\eta,\xi)$ 的等高线图，包括 30 条等高线。用 *Rotate 3D* 分析曲面视图，结果应如图 7.12 和图 7.13 所示。

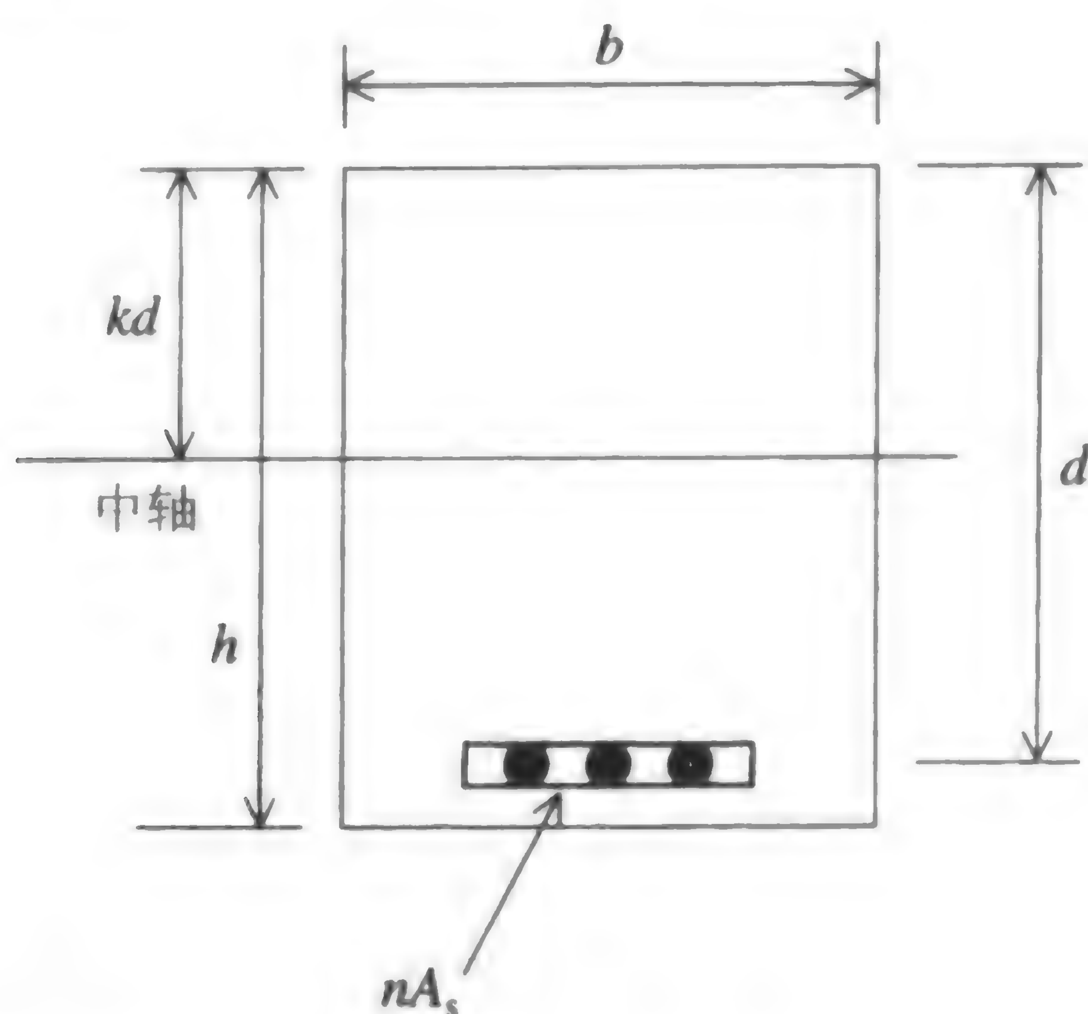


图 7.11　钢筋混凝土梁的剖面图

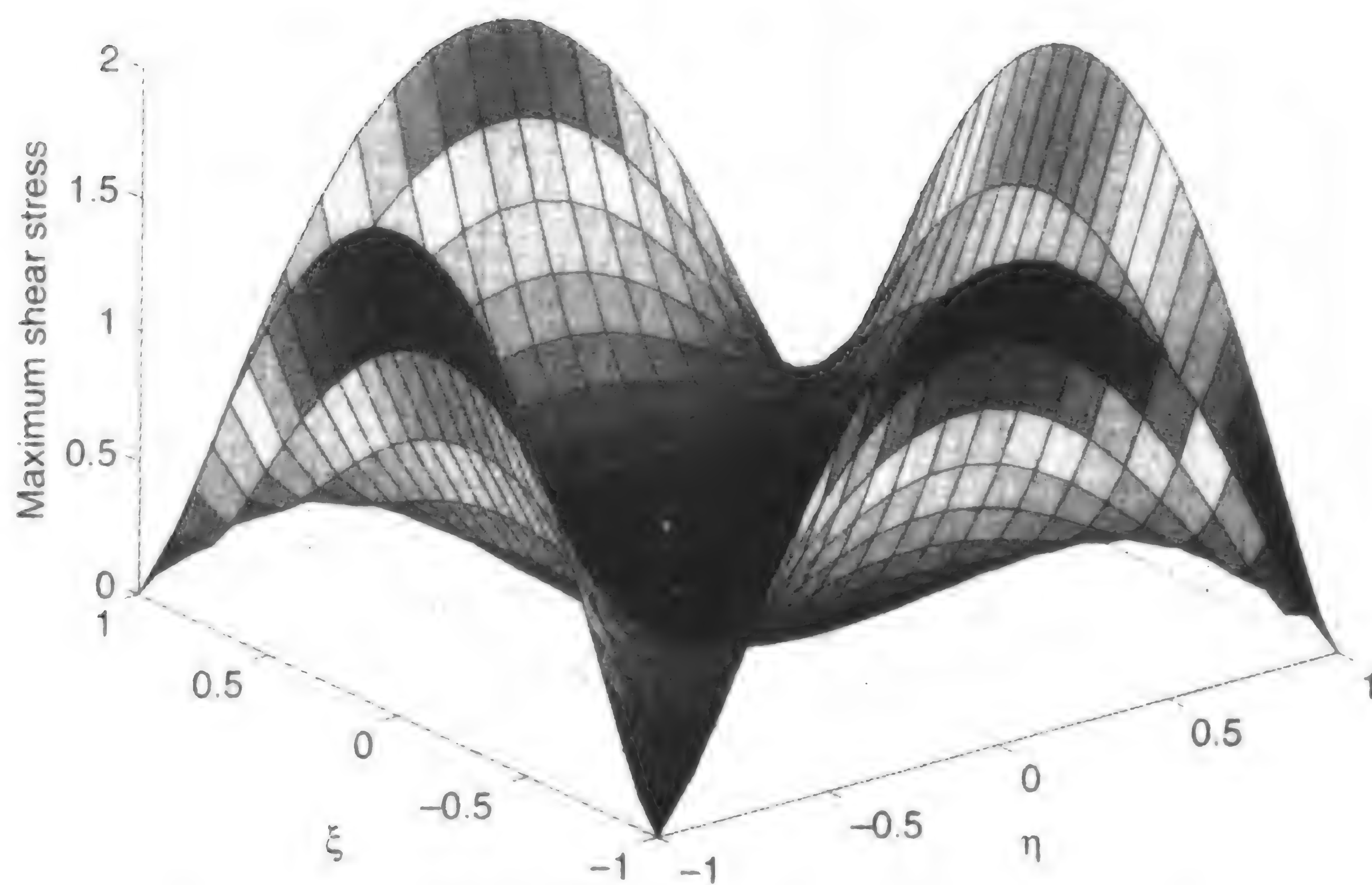


图 7.12　方形横截面梁所受扭力的最大剪切应力平方图

7.10　如表 7.10 所示，表中数据表明当某过程输入量为 x_1 和 x_2 时，输出量是相对于其均值 $z=0$ 的偏差。使用 `stem3` 函数绘制函数图形，得到图 7.14，要求必须使用单独命令画出 $z=0$ 处的平面。使用 `view(-30, 7)` 得到图中所示方向（二维函数为 `stem`）。

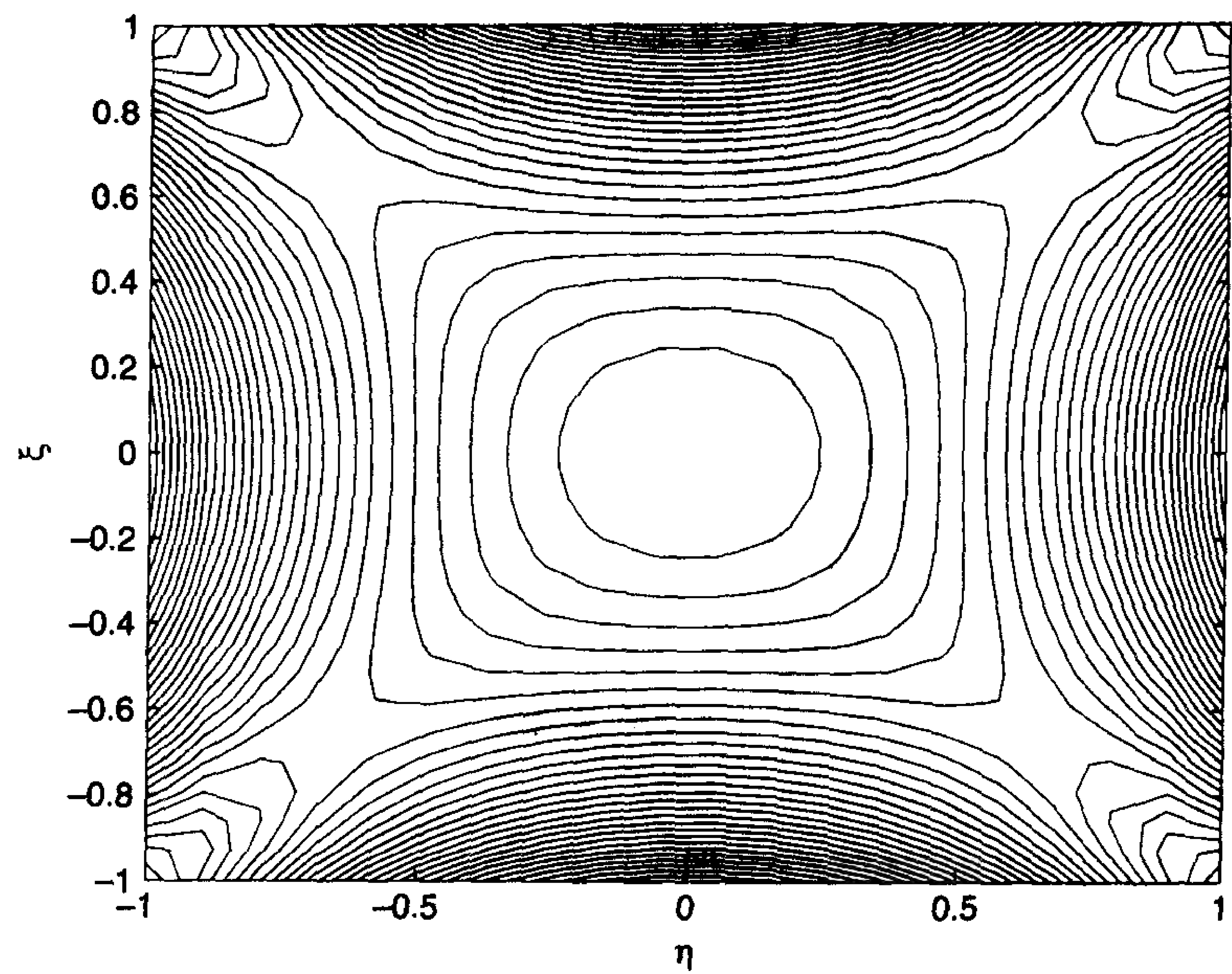


图 7.13 图 7.12 中最大剪切应力的等高线图

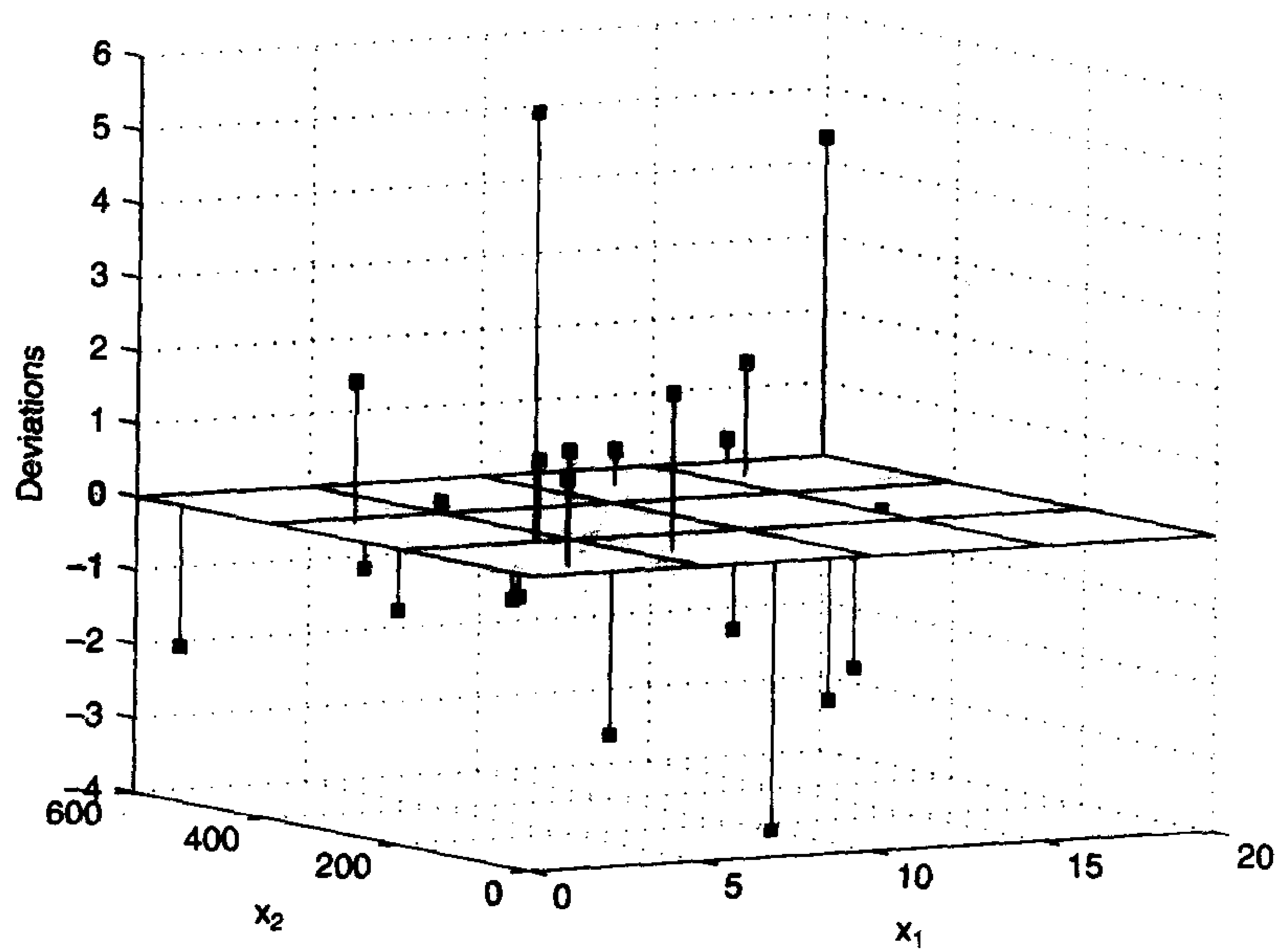


图 7.14 用函数 stem3 得到的以平面 $z=0$ 为基准的偏差

表 7.10 过程标准偏差

x_1	x_2	z	x_1	x_2	z
2	50	1.5713	2	360	−0.6023
8	110	−1.1460	4	205	5.8409
11	120	−2.2041	4	400	−0.3620

（续表）

x_1	x_2	z	x_1	x_2	z
10	550	-1.5968	20	600	4.3341
8	295	-2.8937	1	585	-2.0368
4	200	1.1136	10	540	-1.5415
2	375	1.9297	15	250	0.0302
2	52	1.1962	15	290	-2.1809
9	100	-3.8650	16	510	1.5587
8	300	-0.4763	17	590	0.3222
4	412	-1.3223	6	100	2.1478
11	400	-0.4619	5	400	0.1537
12	500	0.4911			

第8章　机械零件设计

本章介绍不同类型机械零件的各种分析方法。

8.1　向量、受力和刚体平衡

研究向量：

$$\boldsymbol{a} = a_1\boldsymbol{i} + a_2\boldsymbol{j} + a_3\boldsymbol{k}$$

点乘定义为：

$$\boldsymbol{a}\cdot\boldsymbol{a} = a_1^2 + a_2^2 + a_3^2 \tag{8.1}$$

其值由下式获得：

$$a = |\boldsymbol{a}| = \sqrt{\boldsymbol{a}\cdot\boldsymbol{a}} = \sqrt{{a_1}^2 + {a_2}^2 + {a_3}^2} \tag{8.2}$$

如果 $\boldsymbol{a}$ 表示如下：

```
a = [a1,a2,a3]
```

则点乘由下面的函数表示：

```
adot = dot(a,a)
```

其值可以是：

```
maga = sqrt(dot(a,a))
```

或者是：

```
maga = norm(a)
```

$\boldsymbol{a}$ 的方向余弦是：

$$\cos(\alpha_j) = \frac{a_j}{|\boldsymbol{a}|} \quad j = 1,2,3 \tag{8.3}$$

上式为单位向量 $\boldsymbol{u}_a$ 在 $\boldsymbol{a}$ 方向上的分量，可以写为：

$$\boldsymbol{u}_a = \cos\alpha_1\boldsymbol{i} + \cos\alpha_2\boldsymbol{j} + \cos\alpha_3\boldsymbol{k} = \frac{a_1}{|\boldsymbol{a}|}\boldsymbol{i} + \frac{a_2}{|\boldsymbol{a}|}\boldsymbol{j} + \frac{a_3}{|\boldsymbol{a}|}\boldsymbol{k} \tag{8.4}$$

所以，

$$\boldsymbol{a} = |\boldsymbol{a}|\boldsymbol{u}_a \tag{8.5}$$

并且，

$$\boldsymbol{u}_a\cdot\boldsymbol{u}_a = \cos^2\alpha_1 + \cos^2\alpha_2 + \cos^2\alpha_3 = \frac{a_1^2}{|\boldsymbol{a}|^2} + \frac{a_2^2}{|\boldsymbol{a}|^2} + \frac{a_3^2}{|\boldsymbol{a}|^2} = 1$$

其中，

$$\alpha_j = \cos^{-1}\frac{a_j}{|\boldsymbol{a}|} \quad j = 1,2,3 \tag{8.6}$$

在 $\boldsymbol{a}$ 方向上的单位向量 $\boldsymbol{u}_a$ 可表示如下：

```
ua = a/norm(a)
```

方向余弦角度的弧度值是：

```
alpha = acos(a/norm(a))
```

若有另一个向量：

$$\boldsymbol{b} = b_1\boldsymbol{i} + b_2\boldsymbol{j} + b_3\boldsymbol{k}$$

则向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的叉乘定义为：

$$\boldsymbol{c} = \boldsymbol{a}\times\boldsymbol{b} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = (a_2b_3 - a_3b_2)\boldsymbol{i} + (a_1b_3 - a_3b_1)\boldsymbol{j} + (a_1b_2 - a_2b_1)\boldsymbol{k} \tag{8.7}$$

其中，$\boldsymbol{c}$ 垂直于包含 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的平面。叉乘表示为：

```
c = cross(a,b)
```

为了在单位向量 $\boldsymbol{u}_a$ 的方向上确定叉乘的值，使用 3 级标量乘积：

$$c_a = \boldsymbol{u}_a \cdot \boldsymbol{a}\times\boldsymbol{b} \tag{8.8}$$

其中，c_a 是一个标量，则，

```
ca = dot(ua,cross(a,b))
```

这些结论可以用来分析刚体部件的受力和力矩等一系列问题。

例 8.1　力的合成

一个力系如图 8.1 所示。确定合力 $\boldsymbol{F}$ 的值及其方向余弦角 α_i 的大小。

$$\boldsymbol{F} = \boldsymbol{F}_1 + \boldsymbol{F}_2$$

由式(8.6)得：

$$\alpha_i = \cos^{-1}\frac{F_i}{|\boldsymbol{F}|}$$

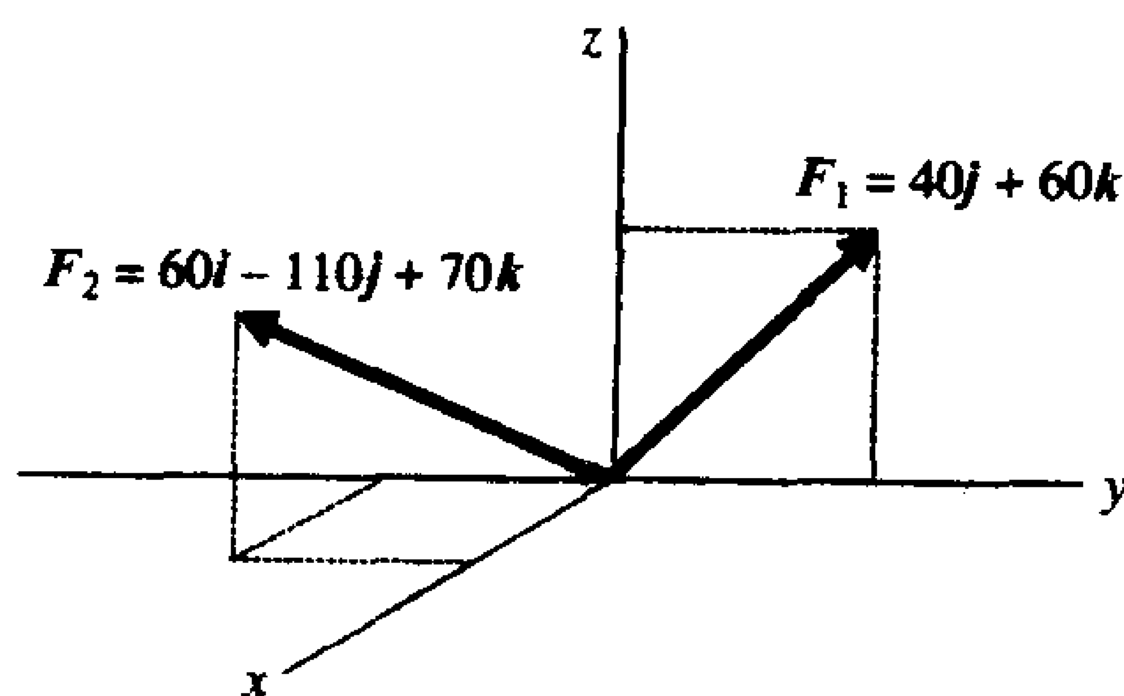


图 8.1　例 8.1 的受力方向

程序如下：

```
F1 = [0,40,60];F2 = [60, - 110,70];
resultantF = norm(F2 + F1)
angles = acos((F2 + F1)/resultantF)' * 180/pi
```

执行程序，得到 $|F| = resultant = 159.3738$, $\alpha = angles = [67.8846, 116.0541, 35.3441]$，用角度表示。

例 8.2　力的分解

一个力如图 8.2 所示。使用式(8.4)和式(8.5)分解作用在 D 点的力 $\boldsymbol{F}$，并用式(8.6)确定该力的方向余弦角 α。程序如下：

```
r = [24, - 16, - 48];
ur = r/norm(r);
F = 30 * ur
angles = acos(ur) * 180/pi
```

执行程序后得到 $\boldsymbol{F} = [12.8571, -8.5714, -25.7143]$, $\alpha = angles = [64.6231, 106.6015, 148.9973]$，用角度表示。

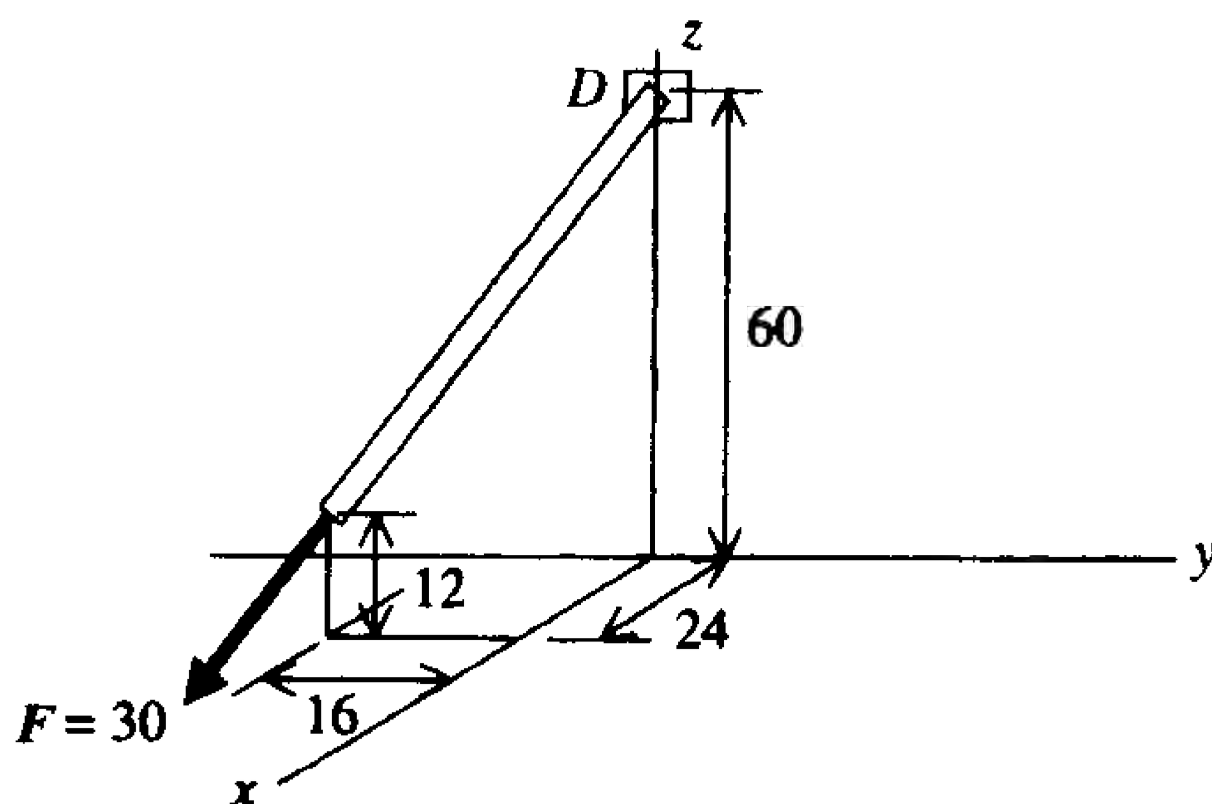


图 8.2　例 8.2 的受力方向

例 8.3　合力的值

一个力系如图 8.3 所示。求 $\boldsymbol{F}_L$ 和 $\boldsymbol{F}_R$ 的合力大小及其分量。

$$\boldsymbol{F} = \boldsymbol{F}_L + \boldsymbol{F}_R$$

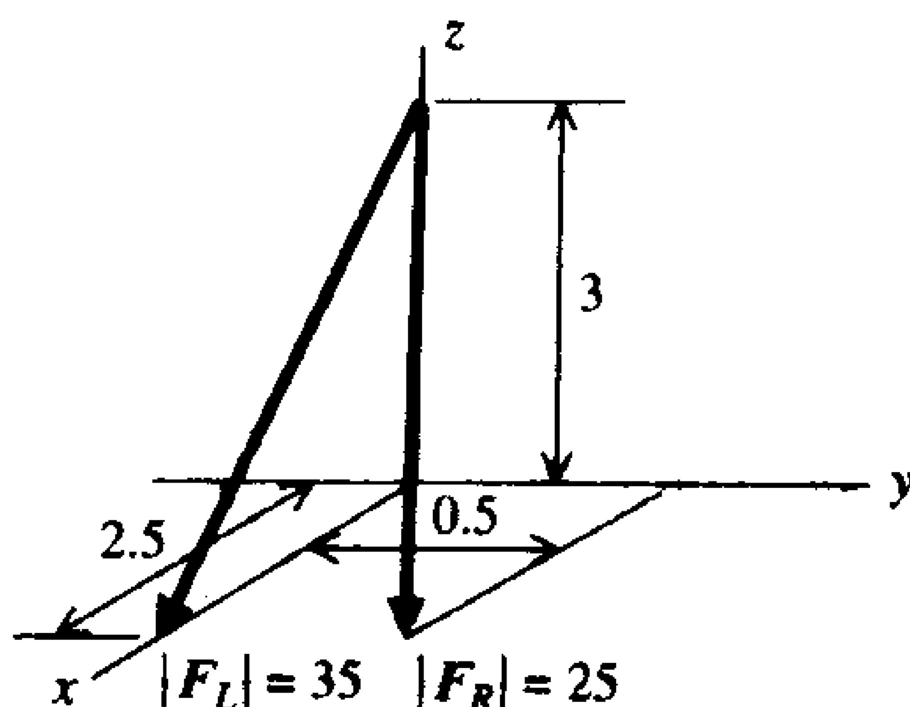


图 8.3　例 8.3 的受力方向

使用式(8.4)、式(8.5)及式(8.2)，程序如下：

```
rl = [2.5,0, - 3];rr = [2.5,0.5, - 3];
F = 35 * rl/norm(rl) + 25 * rr/norm(rr)
```

```
resultant = norm(F)
```

程序执行后得到 $|\boldsymbol{F}| = resultant = 59.8818$。$\boldsymbol{F}$ 分量是 $\boldsymbol{F} = [38.2815, 3.1750, -45.9378]$。

例 8.4　给定方向上的受力分解

分析如图 8.4 所示的力，确定 $\boldsymbol{F}$ 相对于构件 OA 平行方向上的分量（$\boldsymbol{F}_{OA}$）和垂直方向上的分量（$\boldsymbol{F}_{\perp}$），并确定 $\boldsymbol{F}_{\perp}$。使用式(8.4)、式(8.5)和式(8.2)，并记为：

$$\boldsymbol{F} = \boldsymbol{F}_{OA} + \boldsymbol{F}_{\perp}$$

程序是：

```
r = [1,3,1.5];F = [0,125,0];
ur = r/norm(r);
FoaPar = dot(F,ur) * ur
FoaPerp = F-FoaPar
FoaPerpMag = norm(FoaPerp)
```

执行程序后得到：$\boldsymbol{F}_{OA} = FoaPar = [30.6122, 91.8367, 45.9184]$，$\boldsymbol{F}_{\perp} = FoaPerp = [-30.6122, 33.1633, -45.9184]$，$|\boldsymbol{F}_{\perp}| = FoaPerpMag = 64.3848$。

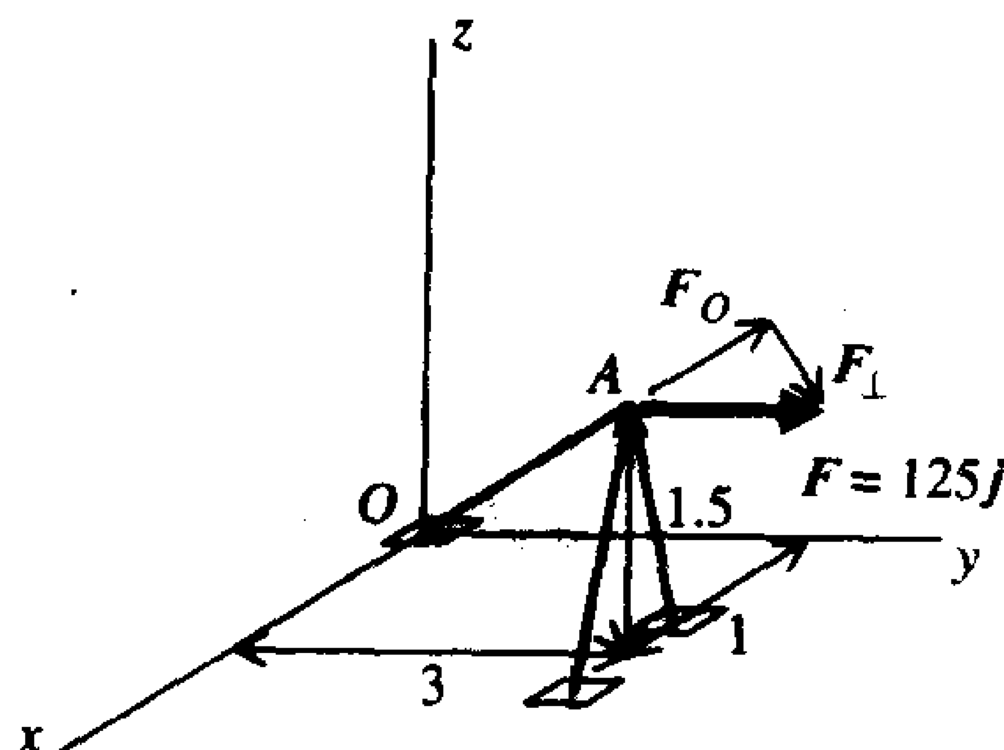


图 8.4　例 8.4 的受力方向

例 8.5　力的平衡

一个力系如图 8.5 所示。确定保持该力系平衡所需的力的分量、大小和方向余弦角 α：

$$\boldsymbol{F} = -(\boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{F}_4)$$

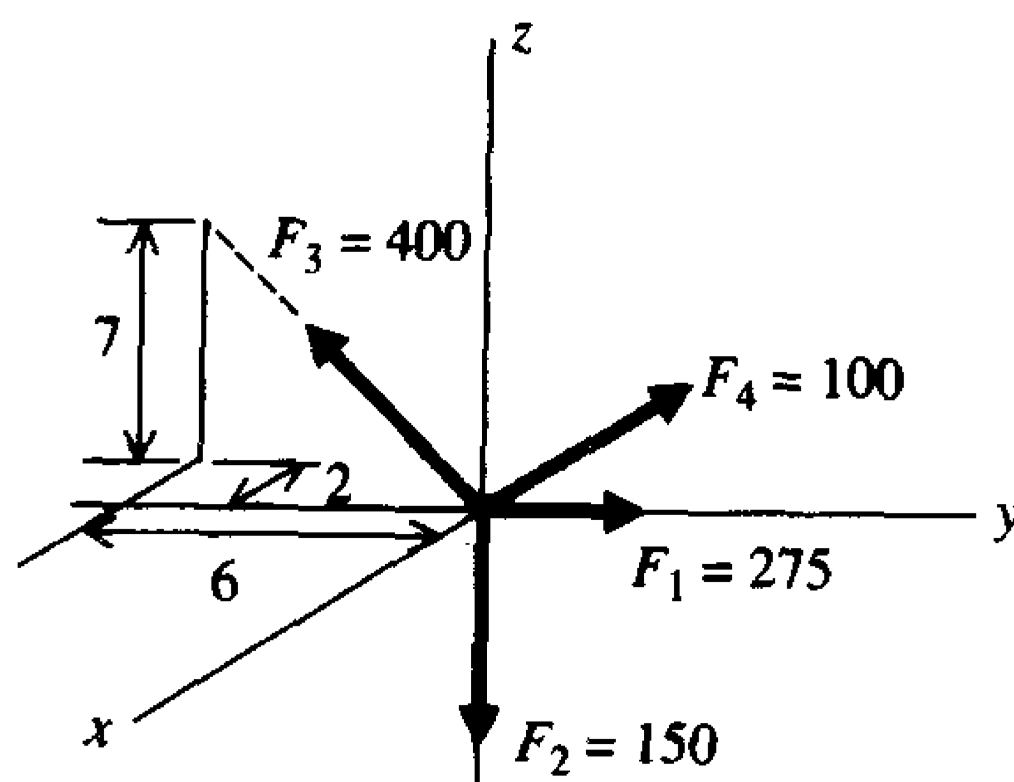


图 8.5　例 8.5 的受力方向

使用式(8.4)、式(8.5)、式(8.2)和式(8.6)，程序如下：

```
F1 = [0,275,0];F2 = [0,0, - 150];
F4 = [ - 100,0,0];r = [ - 2, - 6,7];
F3 = 400 * r/norm(r);
Fequil = - (F1 + F2 + F3 + F4)
FequilMag = norm(Fequil)
angles = acos(Fequil/FequilMag) * 180/pi
```

程序执行后得到：$\boldsymbol{F} = Fequil = [184.7998, -20.6005, -146.7994]$，$|\boldsymbol{F}| = FequilMag = 236.9080$，$\alpha = angles = [38.7350, 94.9885, 128.2904]$，用角度表示。

例 8.6　力的大小

一个力系如图 8.6 所示。当 $\boldsymbol{W} = -85\boldsymbol{k}$ 时，确定 $\boldsymbol{F}_1$，$\boldsymbol{F}_2$ 和 $\boldsymbol{F}_3$ 的值。首先，确定分力 $\boldsymbol{F}_j = |\boldsymbol{F}_j|\boldsymbol{u}_j$。其中，$\boldsymbol{u}_j$ 是在 $\boldsymbol{r}_j = r_{j1}\boldsymbol{i} + r_{j2}\boldsymbol{j} + r_{j3}\boldsymbol{k}$ 方向上的单位向量，$j = 1,2,3$，从式(8.4)和式(8.5)中确定，有下式：

$$\boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{W} = 0$$

或

$$H_1\boldsymbol{i} + H_2\boldsymbol{j} + H_3\boldsymbol{k} = 85\boldsymbol{k}$$

其中，

$$H_n = \sum_{m=1}^{3} |F_m / r_m| r_{nm} \qquad n = 1,2,3$$

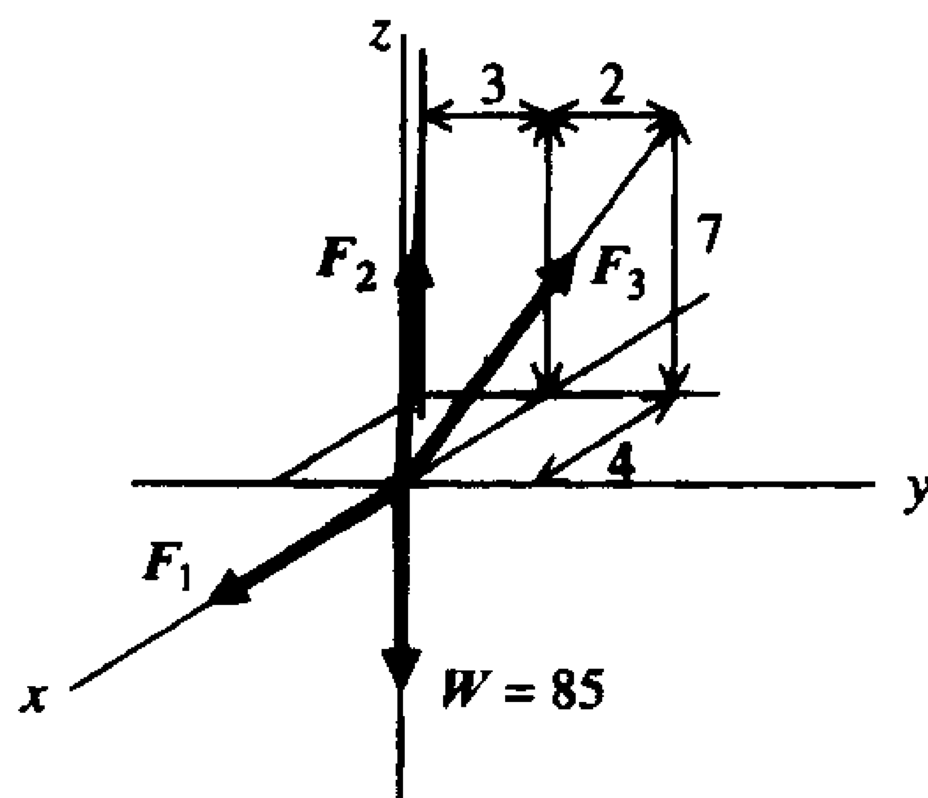


图 8.6　例 8.6 的受力方向

写成矩阵形式如下：

$$\begin{bmatrix} r_{11}/|r_1| & r_{21}/|r_2| & r_{31}/|r_3| \\ r_{12}/|r_1| & r_{22}/|r_2| & r_{32}/|r_3| \\ r_{13}/|r_1| & r_{23}/|r_2| & r_{33}/|r_3| \end{bmatrix} \begin{bmatrix} |F_1| \\ |F_2| \\ |F_3| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 85 \end{bmatrix}$$

程序是：

```
r1 = [1,0,0];r2 = [ - 4, - 3,7];r3 = [ - 4,2,7];
u1 = r1/norm(r1);
u2 = r2/norm(r2);
u3 = r3/norm(r3);
Fmag123 = [u1', u2', u3'] \ [0,0,85]'
```

在这里使用了 2.6.5 节中讨论的左除矩阵方法。

程序执行后得到 $|\boldsymbol{F}_j| = Fmag123 = [48.5714\ 41.7827\ 60.5197]'$；即 $|\boldsymbol{F}_1| = Fmag123(1,1) = 48.5714$，$|\boldsymbol{F}_2| = Fmag123(2,1) = 41.7827$ 和 $|\boldsymbol{F}_3| = Fmag123(3,1) = 60.5197$。

例 8.7 力矩计算

受力如图 8.7 所示。确定 F 作用在构件 OA 上对 O 点的力矩大小。如果 r_a 表示从 O 到 A 的位置向量，r_b 表示从 O 到 B 的位置向量，则 F 在 BA 方向上的分量为 $|F|u_f$。其中，u_f 是 $r_a - r_b$ 的单位向量，则由式(8.7)可得出力矩为：

$$M = r_a \times (|F|u_f)$$

并且其值可以由式(8.2)得到，程序如下：

```
ra = [1.5,3.5,3]; rb = [3.5,5,0];
F = 25 * (ra - rb)/norm(ra - rb);
FaboutA = cross(ra,F);
Mmag = norm(FaboutA)
```

程序执行后得到：$|M| = Mmag = 121.0968$。

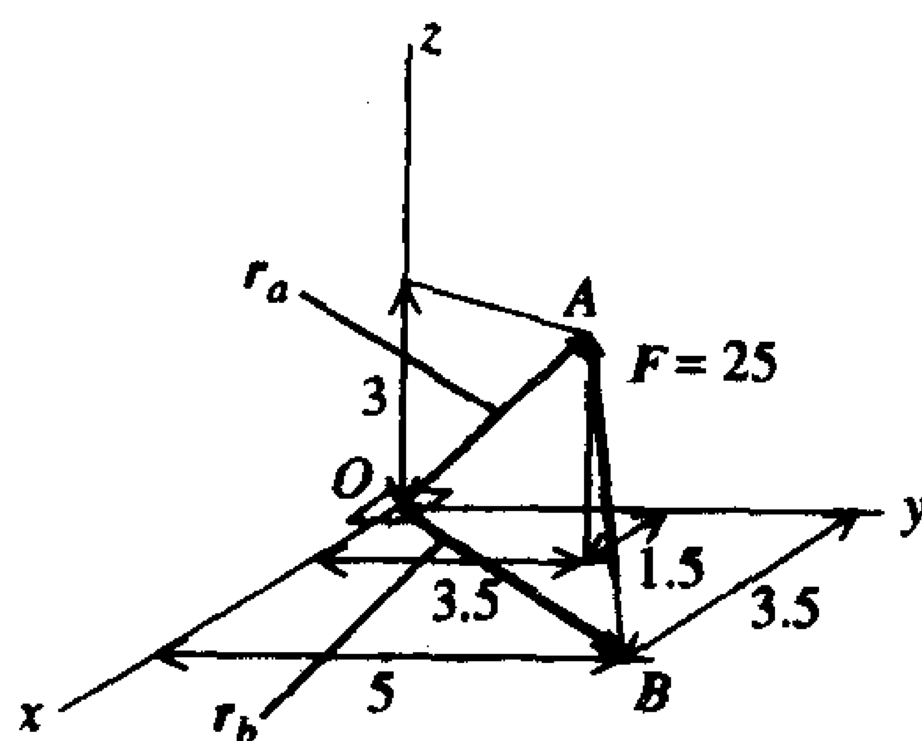


图 8.7 例 8.7 的受力方向

例 8.8 多个力作用下的力矩

作用在一个构件 OAB 上的力系如图 8.8 所示。确定在 O 点由 F_1, F_2, F_3 引起的力矩 M 的值以及方向余弦角 α 的值。这样，

$$M = r_a \times F_1 + r_a \times F_2 + r_b \times F_3$$

其中，r_a 是位置向量 OA，r_b 是位置向量 OB。程序如下：

```
ra = [0,11,0]; rb = [9,11,0];
F1 = [-55,45,30]; F2 = [0,20,0]; F3 = [70,40,-50];
Mro = cross(ra,F1) + cross(ra,F2) + cross(rb,F3);
MroMag = norm(Mro)
angles = acos(Mro/MroMag) * 180/pi
```

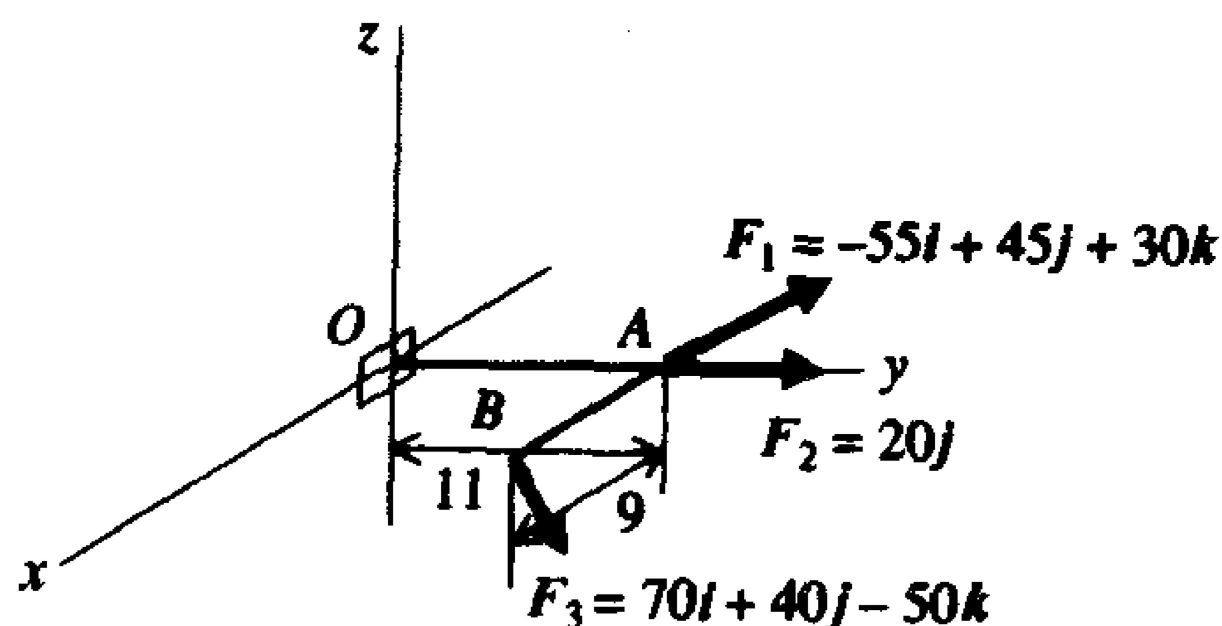


图 8.8 例 8.8 的受力方向

程序执行后得到：$|M| = MroMag = 537.5174$，$\alpha = angles = [114.1602, 33.1563, 68.7290]$，用角度表示。

例 8.9　绕轴转矩

一个力系如图 8.9 所示，其中：

$$\boldsymbol{F} = -30\boldsymbol{i} + 10\boldsymbol{j} - 15\boldsymbol{k}$$

是作用在构件 *ABC* 上的力。在允许的条件下确定力 $\boldsymbol{F}$ 引起的构件 *AB* 绕轴转矩。其中：

$$|\boldsymbol{M}_{AB}| = \boldsymbol{u}_b \cdot \boldsymbol{r}_d \times \boldsymbol{F}$$

和

$$\boldsymbol{M}_{AB} = |\boldsymbol{M}_{AB}|\boldsymbol{u}_b$$

其中，$\boldsymbol{u}_b$ 是 *AB* 方向上的单位向量，$\boldsymbol{r}_d$ 是 *AD* 方向上的向量。程序如下：

```
rb = [0,0,1.5]; rd = [0,1,0]; F = [ -30,10, -15];
ub = rb/norm(rb);
Mab = dot(ub,cross(rd,F))
MabVec = Mab * ub
```

执行程序后得到：$|\boldsymbol{M}_{AB}| = Mab = 30$，$\boldsymbol{M}_{AB} = MabVec = [0,0,30]$。

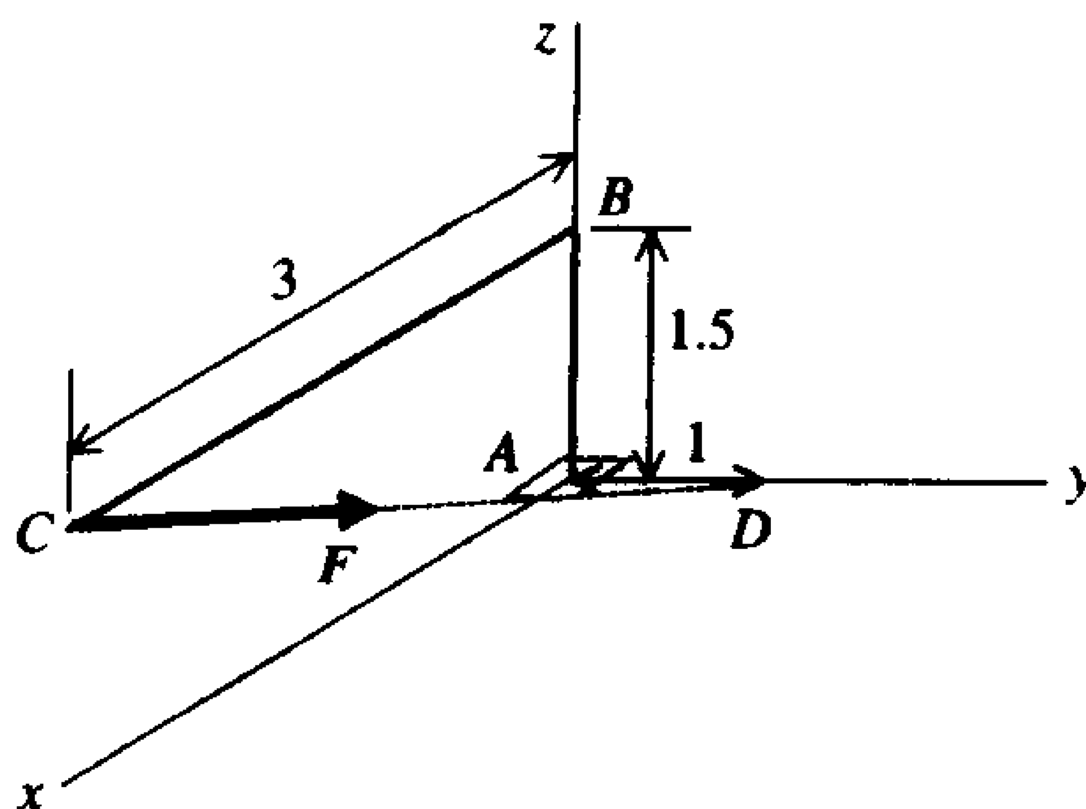


图 8.9　例 8.9 的受力方向

例 8.10　等效力和力矩

一个力系如图 8.10 所示。其中，M 是关于图示轴线的力矩，通过作用在 O 点的等效合力 $\boldsymbol{F}_O$ 和力偶 $\boldsymbol{M}_O$ 代替力系的作用。由此，如果 $\boldsymbol{r}_{AF}$ 是一个表示从 A 到 M 方向上的位置向量($\alpha = 36.87°$表示直角三角形的一个角)，并且 $\boldsymbol{r}_{21} = \boldsymbol{r}_2 - \boldsymbol{r}_1$，则：

$$\boldsymbol{F}_2 = \boldsymbol{F}_2\boldsymbol{r}_{21}/|\boldsymbol{r}_{21}|$$

$$\boldsymbol{M} = \boldsymbol{M}\mathbf{r}_{AF}/|\boldsymbol{r}_{AF}|$$

$$\boldsymbol{F}_0 = \boldsymbol{F}_1 + \boldsymbol{F}_2$$

和

$$\boldsymbol{M}_O = \boldsymbol{M} + \boldsymbol{r}_1 \times \boldsymbol{F}_1 + \boldsymbol{r}_2 \times \boldsymbol{F}_2$$

程序如下：

```
r1 = [0,0,12.5]; r2 = [ -1.75,1.25,12.5];
rAF = [0, -4,3]; F1 = [0,0, -40];
r21 = r2 - r1;
F2 = 15 * r21/norm(r21);
M = 25 * rAF/norm(rAF);
FatO = F1 + F2
MatO = M + cross(r1,F1) + cross(r2,F2)
```

程序执行后得到：$F_O = FatO = [-12.2060, 8.7186, -40.0000]$，$M_O = MatO = [-108.9822, -172.5750, 15.0000]$。

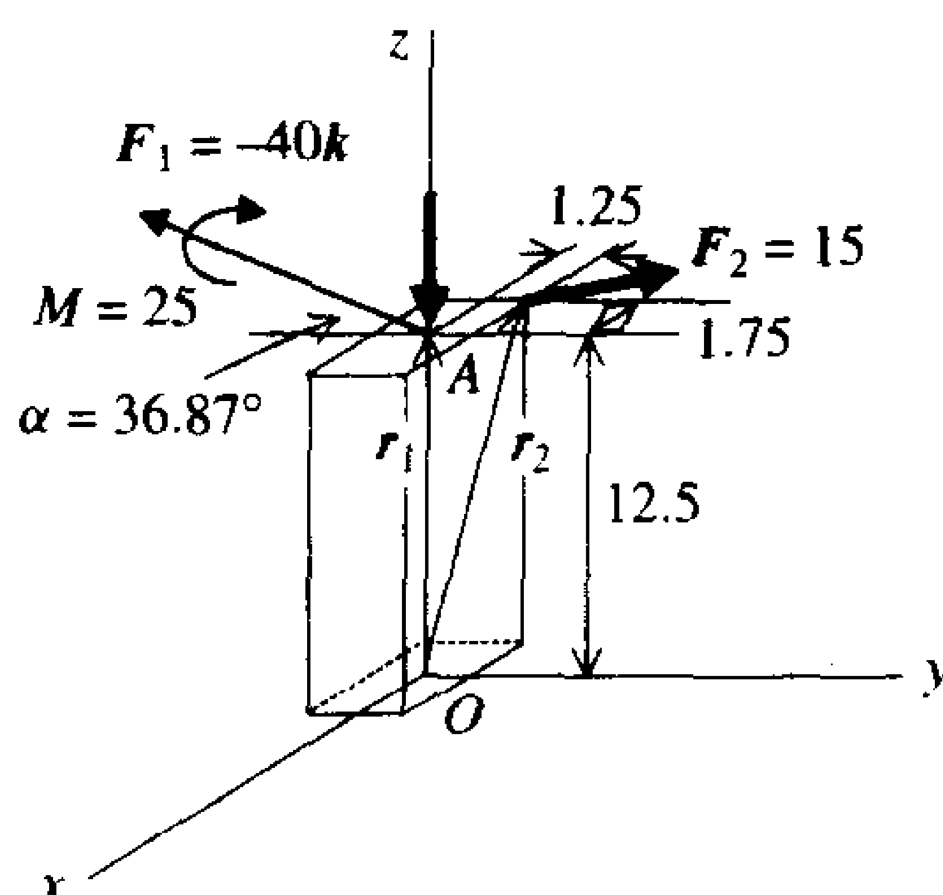


图 8.10　例 8.10 的受力方向

8.2　梁、柱、轴的应力和偏差

8.2.1　静定梁

如果定截面梁横向位移为 $w(x)$，长为 L，弹性模量是 E，横截面的惯性矩是 I，单位长度载荷是 $P_o q(x)$，则无量纲方程如下：

$$\frac{d^4 y}{d\eta^4} = q(\eta) \tag{8.9}$$

其中，$\eta = x/L$，$y = y(\eta) = w/h_o$，$h_o = P_o L^4/EI$，P_o 是常数，表示 q 的最大值。倾角 θ_d，中轴的弯矩 M_d 和剪应力 V_d 用无量纲位移 y 表示为：

$$\begin{aligned} \theta &= \frac{\theta_d}{P_o L^3/EI} = \frac{dy}{d\eta} \\ M &= \frac{M_d}{P_o L^2} = \frac{d^2 y}{d\eta^2} \\ V &= \frac{V_d}{P_o L} = \frac{d^3 y}{d\eta^3} \end{aligned} \tag{8.10}$$

使用无量纲量能得到数值解。这些方法明显地与梁的几何形状、物理特性和载荷的大小无关，而与边界条件和梁的载荷形式有关。

梁中最大的剪应力和弯曲应力分别是：

$$\tau = \frac{\alpha V}{A} \quad \text{和} \quad \sigma = \frac{Mc}{I}$$

其中，A 是梁的横截面面积，α 是一个关于梁的横截面形状的系数，c 是中性轴到梁横截面顶端（如果不对称，则为底端）边缘的距离。横截面是实心圆时，$\alpha = 4/3$；横截面是矩形时，$\alpha = 3/2$；横截面是圆环时，$\alpha = 2$。

除了为式(8.9)获取一个通用的分析方法，用来对不同边界与载荷情况下问题的求解，还可以使用 bvp4c 这种数值解法。这个函数在 5.5.4 节中讨论过。另外，可以将这个结论推广

到例 5.15 中。

为了将式(8.9)转换成 bvp4c 允许的形式,将式(8.9)重新表示为 4 个一阶方程组,即:

$$y_1 = y \qquad y_2 = \frac{dy}{d\eta}$$
$$y_3 = \frac{d^2 y}{d\eta^2} \qquad y_4 = \frac{d^3 y}{d\eta^3} \tag{8.11}$$

则式(8.9)变成:

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_2}{d\eta} = y_3$$
$$\frac{dy_3}{d\eta} = y_4 \qquad \frac{dy_4}{d\eta} = q \tag{8.12}$$

从式(8.10)和式(8.11)中看出,y_2 是转角,y_3 是无量纲力矩,y_4 是无量纲剪应力。

每一根梁的末端都有四种混合类型:简支端、固定端、自由端和左端受应用力矩 M_l,右端受应用力矩 M_r 作用下的梁。考虑这些梁在载荷作用下的结果。根据式(8.10),这些边界条件的数学表达式在表 8.1 中给出。

表 8.1 梁的四种常用边界条件

类型	边界条件	bvp4c 边界条件	条件值
简支端($y = M = 0$)	$y = d^2y/d\eta^2 = 0$	$\eta = 0,1: y_1 = 0, y_3 = 0$	1
固定端($y = \theta = 0$)	$y = dy/d\eta = 0$	$\eta = 0,1: y_1 = 0, y_2 = 0$	2
自由端($M = V = 0$)	$d^2y/d\eta^2 = d^3y/d\eta^3 = 0$	$\eta = 0,1: y_3 = 0, y_4 = 0$	3
简支端加外部力矩	$y = 0$	$\eta = 0: y_1 = 0, y_3 = -M_l$	4
($y = 0; M = -M_l$ 或 $+M_r$)	$d^2/d\eta^2 = -M_l$ 或 $+M_r$	$\eta = 1: y_1 = 0, y_3 = -M_r$	

为了具有载荷类型的灵活性,考虑集中载荷和三角载荷两种类型。作为特例,三角载荷是均匀载荷。载荷类型如图 8.11 所示。假设集中载荷在其应用点上的较小区域内为常量,若集中应力作用在 e_1,则 q 可以表示为:

$$q = \frac{1}{0.01}[u(\eta - e_1 + 0.005) - u(\eta - e_1 - 0.005)]$$

其中,$u(\eta)$是单位阶跃函数,可以任意设置载荷到梁 1%处的宽度。三角载荷可表示为:

$$q = \frac{q_l - q_r}{e_l - e_r}\eta + \frac{q_r e_l - q_l e_r}{e_l - e_r}$$

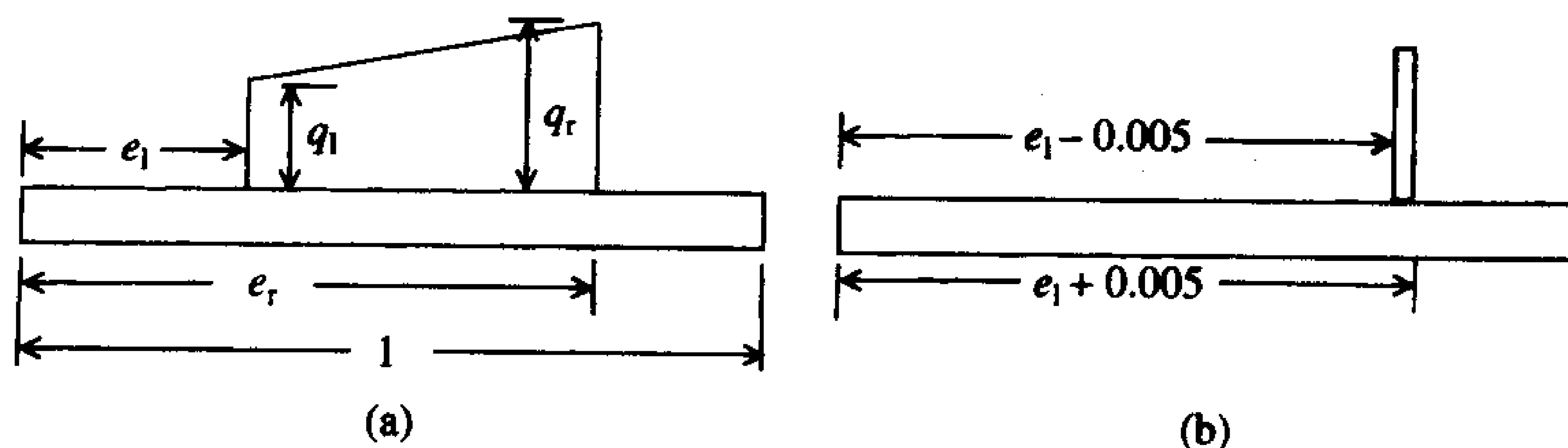


图 8.11 (a) 梁上普通载荷说明;(b) 集中载荷的近似表示

q_l 和 q_r 的取值范围为 0 和 1 之间,两个值相互间的大小不必加以考虑,但必须保证一个值为 1 或两个值都为 1。当两个值都为 1 时,则是一个均匀载荷,载荷位置是 $e_r > e_l$。

依照例 5.16 的结论，首先创建函数文件 **StaticBeamDSMV**，然后，在表 8.1 给出的边界条件下，计算图 8.11 中普通载荷和三角载荷作用下式(8.9)的值。根据表 8.1 中的条件值选择每根梁末端的边界条件。$k_l=2,3,4$ 表示左端，$k_r=2,3,4$ 表示右端。将 M_l 或 M_r 设为 0，从 case 4 获得 case 1。为了使程序简单，将 case 4 和 case 1 合并，统称为 case 1。仅当考虑 M_l 或 M_r 的影响时，用“Point”，“Triangular”或“None”分别说明不同的载荷情况。

```
function[y,eta] = StaticBeamDSMV
global Ltype el ql er qr kl kr Ml Mr
solinit = bvpinit(linspace(0,1,10),[0.5,0.5,0.5,0.5]);
beamsol = bvp4c(@BeamODEq@BeamBC1,solinit,[]);
eta = linspace(0,1,200);
y = deval(beamsol,eta);

function bc = BeamBC1(y0,y1)
global Ltype el ql er qr kl kr Ml Mr
switch kl
  case 1
  bc = [y0(1);y0(3) + Ml];
  case 2
  bc = [y0(1);y0(2)];
  case 3
  bc = [y0(3) + Ml;y0(4)];
end
switch kr
  case 1
  bc = [bc;y1(1);y1(3) - Mr];
  case 2
  bc = [bc;y1(1);y1(2)];
  case 3
  bc = [bc;y1(3) - Mr;yl(4)];
end

function dydx = BeamODEq(x,y)
global Ltype el ql er qr kl kr Ml Mr
switch Ltype
  case 'Point'
    q = (x >= el - 0.005&x <= el + 0.005)/0.01;
  case 'Triangular'
    q = ((ql - qr)/(el - er) * x + (qr * el - ql * er)/(el - er)). * (x >= el&x <= er);
  case 'None'
    q = 0;
end
dydx = [y(2);y(3);y(4); - q];
```

使用 subplot 命令可以将位移、转角、力矩和剪应力画在一张图中。另外，通过调用 **BeamDisplay** 函数还可以将载荷类型、边界条件，以及位移、转角、力矩和剪应力的最大值和最小值显示在命令窗口中：

```
function BeamDisplay(y,eta)
global Ltype el ql er qr kl kr Ml Mr
Bc = char('Hinged','Clamped','Free');
```

```
ylab = char('Displacement','Slope','Moniment','Shear');
disp('')
switch Ltype
  case'Point'
    disp(['Load type;'Ltype'Applied at eta = 'num2str(el)])
  case'Triangular'
    if ql == qr & ql ~= 0
    disp(['Load type:Uniform Applied from eta = 'num2str(el)'to eta = '
            num2str(er)])
  else
    disp(['Load type:'Ltype'Applied from eta = 'num2str(el)'to eta = '
            num2str(er)])
    disp(['ql = 'num2str(ql)'at eta = 'num2str(el)'and qr = 'num2str(qr)
            'at eta = 'num2str(er)])
    end
  case 'None'
    disp(['Load type:None'])
end
MMl = [];MMr = [];
if kl == 1
  MMl = ['(Ml = 'num2str(Ml)')'];
end
if kr == 1
  MMr = ['(Mr = 'num2str(Mr)')'];
end
disp('')
disp(['Boundary conditions:'])
disp(['Left end:'Bc(kl,:)MMl])
disp(['Right end:'Bc(kr,:)MMr])
disp('')
for k = 1:4
  [wmax,indx] = max(y(k,:));
  disp(['Max'ylab(k,:)' = 'num2str(wmax)'at eta = 'num2str(eta(indx))])
  [wmax,indx] = min(y(k,:));
  disp(['Min'ylab(k,:)' = 'num2str(wmin)'at eta = 'num2str(eta(indx))])
  disp('')
  subplot(2,2,k)
  plot(eta,y(k,:),'k-')
  xlable('\eta')
  ylable(ylab(k,:))
end
```

以下通过几个例子说明这两个函数的使用。

例8.11 均布载荷作用的简支梁

分析一个均布载荷作用的简支梁。梁的位移、倾角、动量和剪应力的分布情况用以下程序来获得。

```
global Ltype el ql er qr kl kr Ml Mr
Ltype = 'Triangular';
kl = 1,kr = 1;
el = 0;er = 1;
```

```
ql = 1; qr = 1;
Ml = 0; Mr = 0;
[y2 eta] = StaticBeamDSMV;
BeamDisplay(y2, eta);
```

执行程序后的结果如图 8.12 所示，在 MATLAB 命令窗口中显示下面的信息：

```
Load type: Uniform Applied from eta = 0 to eta = 1

Boundary conditions:
  Left end: Hinged (Ml = 0)
  Right end: Hinged (Mr = 0)

Max Displacement = 0 at eta = 0
Min Displacement = -0.01302 at eta = 0.49749

Max Slope    = 0.041667 at eta = 1
Min Slope    = -0.041667 at eta = 0

Max Monment     = 0.125 at eta = 0.49749
Min Monent      = 0 at eta = 1

Max Shear    = 0.5 at eta = 0
Min Shear    = -0.5 at eta = 1
```

该问题采用解析方法[①] 得到和上面相同的结果。

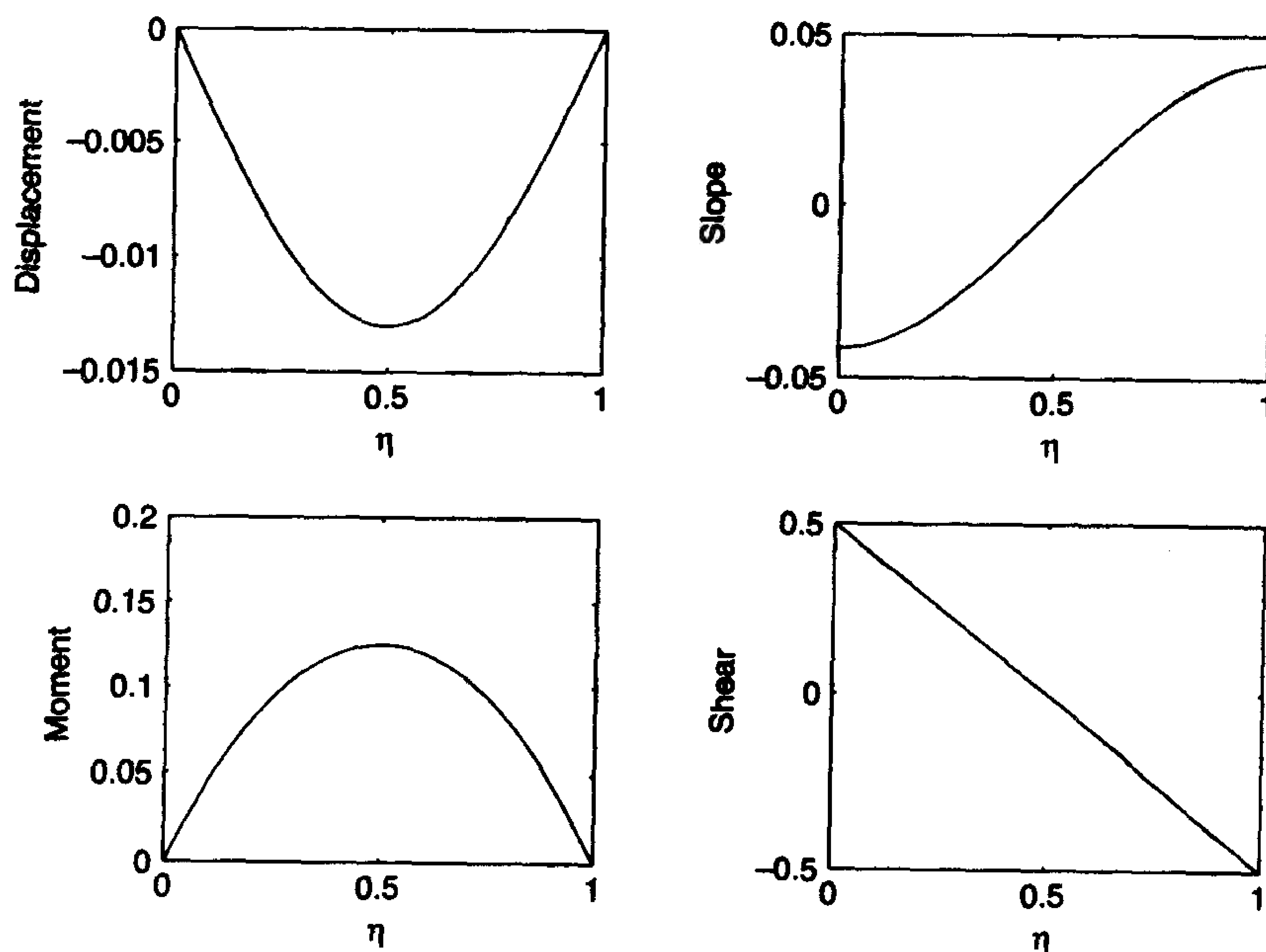


图 8.12　沿长度均布载荷作用的简支梁

① 参见 R. L. Norton, *Design of Machinery*, McGraw-Hill, New York, 1992 年，第 1003 页。

例 8.12　集中载荷作用的简支梁

研究一个两端支撑的简支梁，在 $\eta=0.5$ 处受集中载荷作用。程序如下：

```
global Ltype el ql er qr kl kr Ml Mr
Ltype = 'Point';
kl = 1;kr = 1;
el = 0.5;er = 0.5;
ql = 1;qr = 1;
Ml = 0;Mr = 0;
[y2 eta] = StaticBeamDSMV;
BeamDisplay(y2,eta);
```

执行程序后的结果如图 8.13 所示，在 MATLAB 命令窗口中显示下面的信息：

```
Load type:Point Applied at eta = 0.5

Boundary conditions:
  Left end:Hinged (Ml = 0)
  Right end:Hinged (Mr = 0)

Max Displacement = 0 at eta = 0
Min Displacement = -0.020829 at eta = 0.50251

Max Slope     = 0.06249 at eta = 1
Min Slope     = -0.06249 at eta = 0

Max Moment    = 0.2484 at eta = 0.50251
Min Moment    = 0 at eta = 0

Max Shear     = 0.49994 at eta = 0
Min Shear     = -0.4994 at eta = 0.50754
```

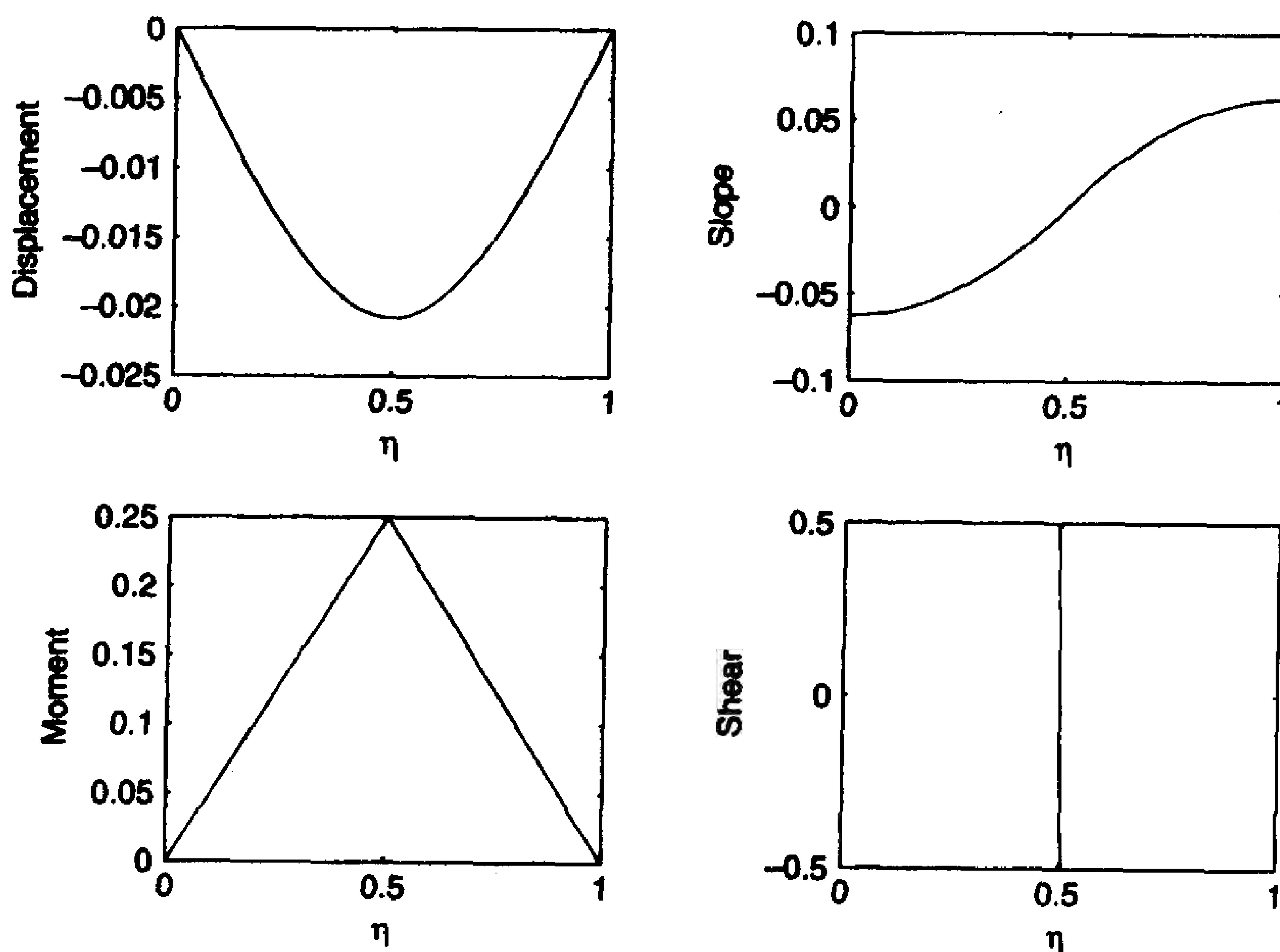


图 8.13　在 $\eta=0.5$ 处有集中载荷作用的简支梁

该问题用解析方法得到[①]：$y_{max}(\eta=0.5)=0.0208$，$\theta_{max}(\eta=1)=0.0625$，$M_{max}(\eta=0.5)=0.250$，$V_{max}(\eta=0)=0.500$。

例 8.13　均布载荷作用的悬臂梁

如果梁在 $\eta=0$ 处固定，在 $\eta=1$ 处悬空，则这种梁称为悬臂梁。此梁受 $\eta=0.5$ 至 $\eta=1$ 区间内的均布载荷作用。程序如下：

```
global Ltype el ql er qr kl kr Ml Mr
Ltype = 'Triangular';
kl = 2; kr = 3;
el = 0.5; er = 1.0;
ql = 1; qr = 1;
Ml = 0; Mr = 0;
[y2 eta] = StaticBeamDSMV;
BeamDisplay(y2, eta);
```

执行程序后的结果如图 8.14 所示，在 MATLAB 命令窗口中显示下面的信息：

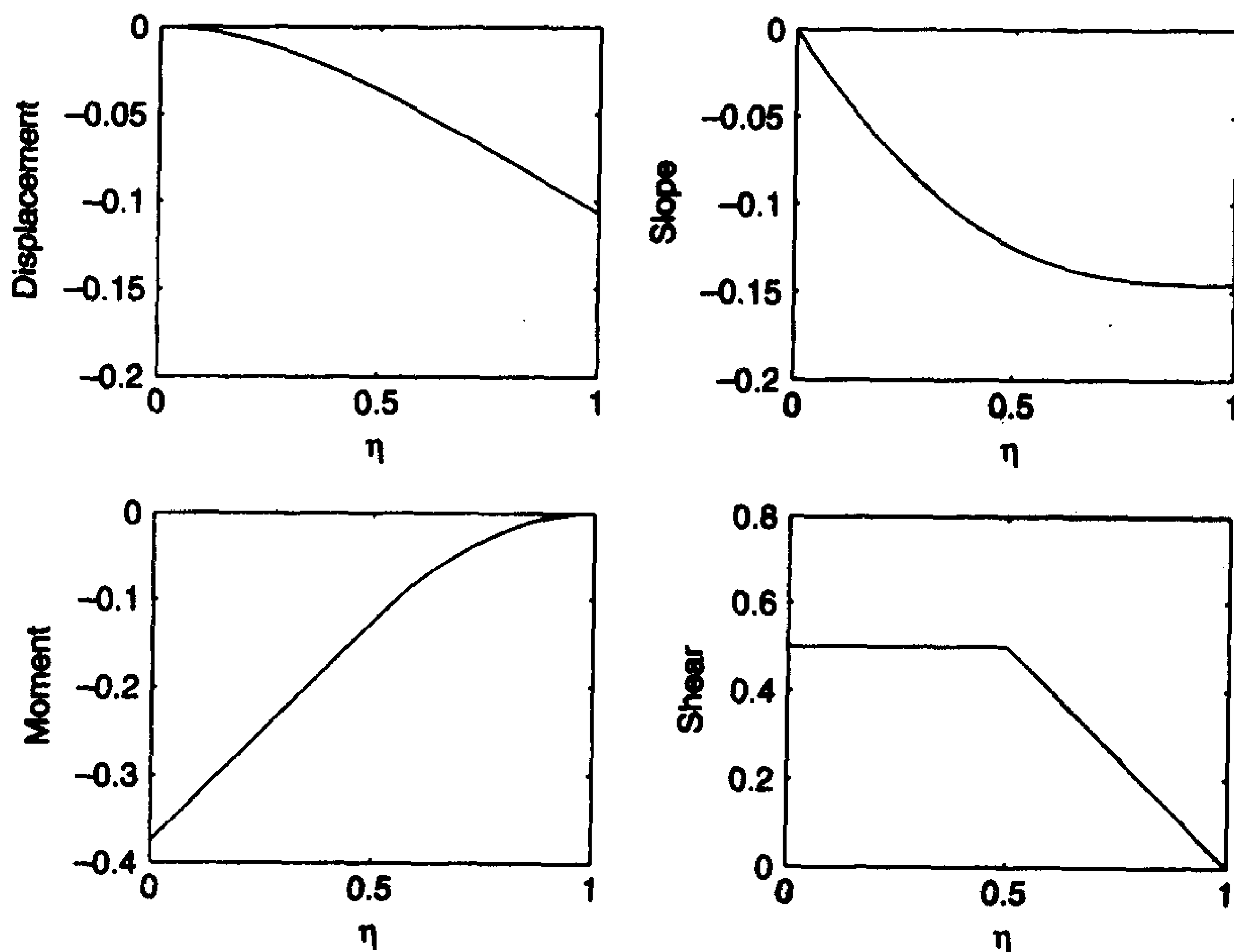


图 8.14　$\eta=0.5$ 至 $\eta=1$ 范围内受均布载荷作用的悬臂梁

Load type: Uniform Applied from eta = 0.5 to eta = 1

Boundary conditions:
  Left end: Clamped
  Right end: Free

Max Displacement = 0 at eta = 0
Min Displacement = −0.10677 at eta = 1

---

① 参见 R. L. Norton，出处同上，第 1003 页。

```
Max Slope      =0 at eta=0
Min Slope     = -0.14583 at eta=1

Max Moment     =0 at eta=1
Min Moment     = -0.375 at eta=0

Max Shear     =0.5 at eta=0
Min Shear     =0 at eta=1
```

该问题用解析方法得到[①]: $y_{max}(\eta=1)=0.1068$, $\theta_{max}(\eta=1)=0.1458$, $M_{max}(\eta=0)=0.375$, $V_{max}(\eta=0)=0.500$。

例 8.14 集中载荷作用的悬臂梁

分析一悬臂梁,在 $\eta=0$ 处固定,在 $\eta=1$ 处悬空,集中载荷作用在 $\eta=0.5$ 处。程序为:

```
global Ltype el ql er qr kl kr Ml Mr
Ltype = 'Point';
kl = 2; kr = 3;
el = 0.5; er = 0.5;
ql = 1; qr = 1;
Ml = 0; Mr = 0;
[y2 eta] = StaticBeamDSMV;
BeamDisplay(y2, eta);
```

执行程序后的结果如图 8.15 所示,在 MATLAB 命令窗口中显示下面的信息:

```
Load type: Point Applied at eta=0.5

Boundary conditions:
  Left end: Clamped
  Right end: Free

Max Displacement=0 at eta=0
Min Displacement= -0.10416 at eta=1

Max Slope      =0 at eta=0
Min Slope     = -0.12499 at eta=0.50754

Max Moment      =0 at eta=0.50754
Min Moment     = -0.49994 at eta=0

Max Shear      =0.99988 at eta=0
Min Shear     =0 at eta=0.50754
```

对这个问题用解析方法[②]得: $y_{max}(\eta=1)=0.1042$, $\theta_{max}(\eta=0.5$ 到 $\eta=0.5)=0.1250$, $M_{max}(\eta=0)=0.500$ 和 $V_{max}(\eta=0)=1.000$。

---

① 参见 R.L.Norton,出处同上,第 1002 页。

② 参见 R.L.Norton,出处同上,第 1002 页。

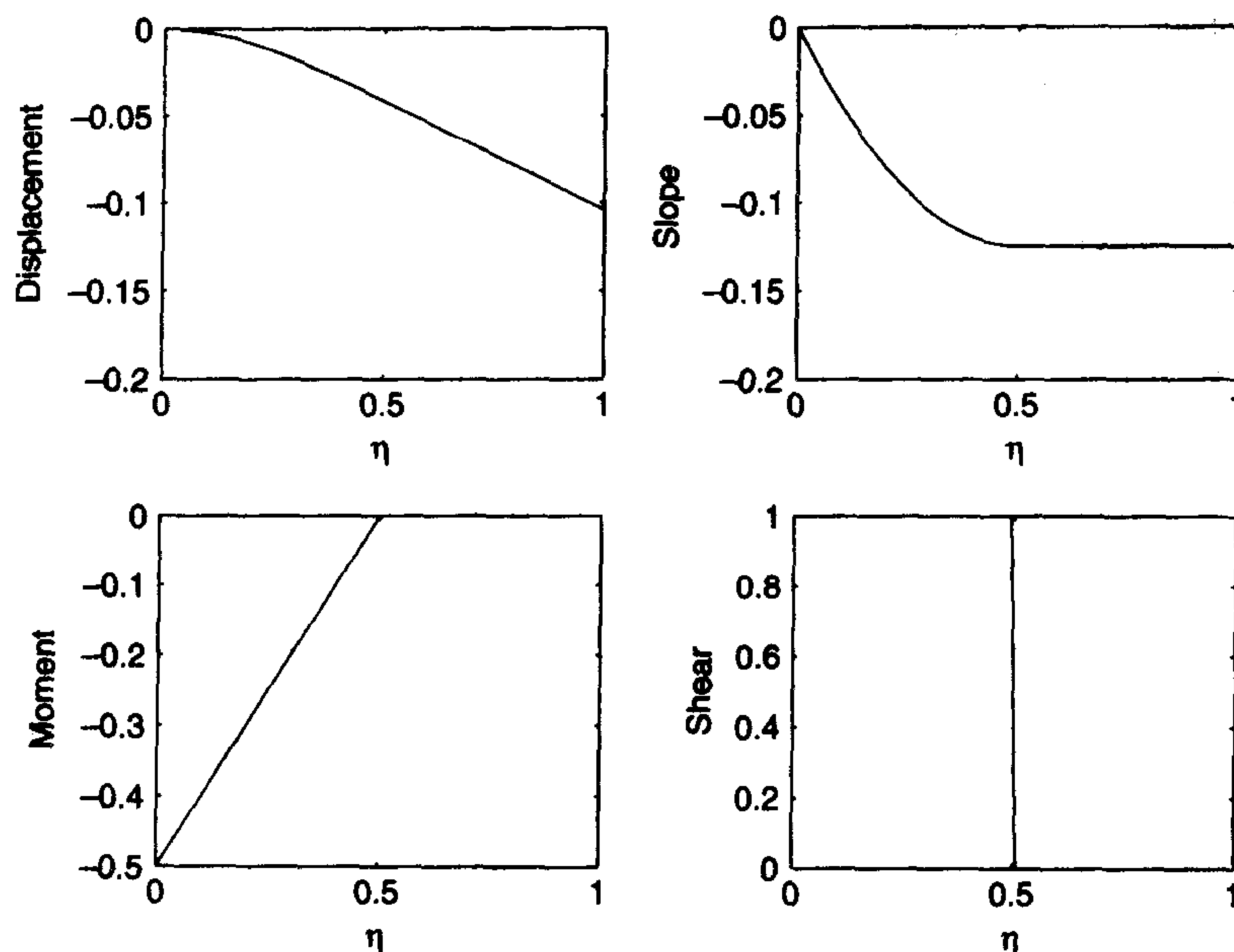


图 8.15　在 $\eta=0.5$ 处受集中载荷作用的悬臂梁

例 8.15　三角载荷作用的悬臂梁

考虑一根梁，固定端在 $\eta=0$ 处，自由端在 $\eta=1$ 处。三角载荷作用在 $\eta=0$ 处，$q_l=0$；作用在 $\eta=1$ 处，$q_r=1$。程序如下：

```
global Ltype el ql er qr kl kr Ml Mr
Ltype = 'Triangular';
kl = 2;kr = 3;
el = 0;er = 1;
ql = 0;qr = 1;
Ml = 0;Mr = 0;
[y2 eta] = StaticBeamDSMV;
BeamDisplay(y2,eta);
```

执行程序后的结果如图 8.16 所示，在 MATLAB 命令窗口中显示下面的信息：

```
Load type:Triangular Applied from eta = 0 to eta = 1
ql = 0 at eta = 0 and qr = 1 at eta = 1

Boundary conditions:
  Left end:Clamped
  Right end:Free

Max Displacement = 0 at eta = 0
Min Displacement = -0.091666 at eta = 1

Max Slope      = 0 at eta = 0
Min Slope      = -0.125 at eta = 1

Max Moment     = 0 at eta = 1
Min monent     = -0.33333 at eta = 0
```

```
Max Shear     = 0.5 at eta = 0
Min Shear     = 0 at eta = 1
```

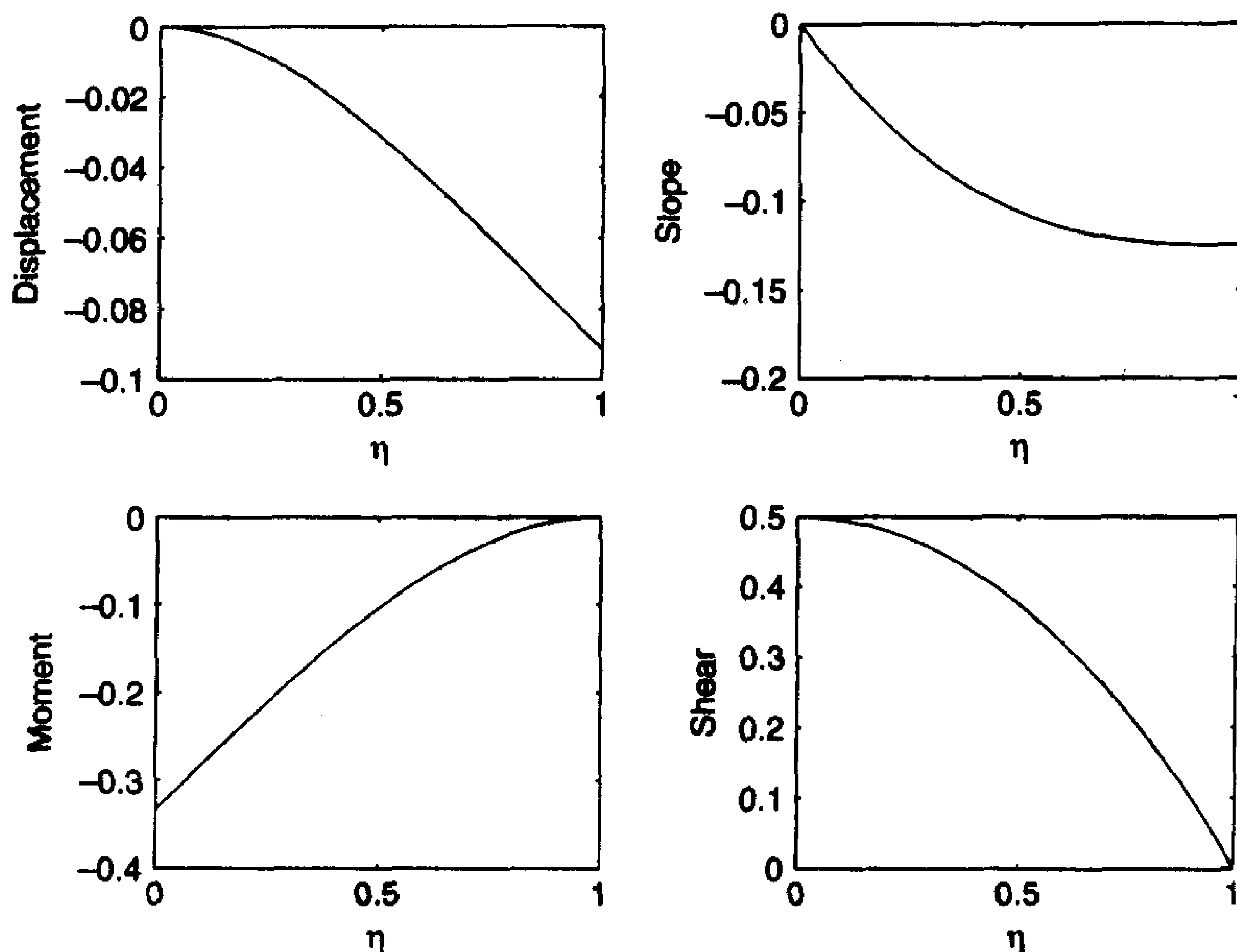


图 8.16 在 $\eta=0$ 处 $q_1=0$,在 $\eta=1$ 处 $q_r=1$,三角载荷作用下的悬臂梁

该问题用解析方法① 得到:

$y_{max}(\eta=1) = -11/120 = -0.091\,66$, $\theta_{max}(\eta=1) = -0.125$, $M_{max}(\eta=0) = -1/3 = 0.333$,和 $V_{max}(\eta=0) = 0.5$。

8.2.2 柱的屈曲

研究一个长度为 L,横截面积为 A,弹性模量为 E,惯量矩为 I,屈服强度为 S_y 的构造柱。受到轴向压缩载荷 P,该载荷和柱轴同心并且通过 A 的质心。临界载荷 P 的值记为 P_{cr},用下式估算:

$S_r > \pi\sqrt{2E/S_y}$:

$$P_{cr} = \frac{\pi^2 AE}{S_r^2}$$

$S_r \leqslant \pi\sqrt{2E/S_y}$:

$$P_{cr} = A\left[S_y - \frac{1}{E}\left(\frac{S_y S_r}{2\pi}\right)^2\right]$$

其中,$S_r = L_{eff}/k$,$k=\sqrt{I/A}$是横截面的惯性半径,对悬臂压杆 $L_{eff}=2.1L$;对两端铰接柱 $L_{eff}=L$;对一端固定,一端铰接柱 $L_{eff}=0.8L$;对双端固定柱 $L_{eff}=0.65L$。

对于一个实心矩形横截面,$k = h/2\sqrt{3}$,其中 h 是横截面的高度;对于一个实心圆形横截

① 参见 R. J. Roark 和 W. C. Young, *Formulas for Stress and Strain*, McGraw-Hill, New York, NY, 第 98 页。

面，$k=r/2$，其中 r 是圆的半径；对于一个内径 r_i 和外径 r_o 的圆环，为：

$$k=0.5\sqrt{r_o^2+r_i^2}$$

或者，如果 $r_o=r_i+t$，则：

$$k=0.5\sqrt{r_o^2+(r_o-t)^2}$$

如果使用安全系数 F_s，则 $P=P_{cr}/F_s$。

当 P 是偏移量为 e 的偏心载荷时，则：

$$P_{cr}=AS_y\left[1+\frac{ec}{k^2}\sec\left(\frac{L_{eff}}{k}\sqrt{\frac{P_{cr}}{4AE}}\right)\right]^{-1}$$

其中，c 是从截面 A 的形心到截面 A 的外周的距离。当 $ec/k^2 \leqslant 0.025$ 时，该关系有效。当截面是圆周时，$c=r_o$。P_{cr}在等式的两端出现，所以必须使用 fzero 求解。

下面通过两个例子来说明这些关系。

例 8.16　确定圆柱直径

一个 13 ft 的悬臂钢圆柱受 150 000 lb 的轴向压力载荷作用。如果圆柱的截面是壁厚为 0.75 in 的空心圆，那么当 $E=3\times10^7$psi，$S_y=55\,000$ psi 和 $F_s=3$ 时，确定最小外直径 d。使用 fzero 来获得 d 的估计值。首先，创建一个函数 **ColumnBuckling**。程序如下：

```
function ColumnDiameter
opt = optimset('display','off');
Leff = 2.1 * 12 * 13; P = 150000; Sy = 55000;
E = 3e + 7; Fs = 3; t = 0.75;
d = 2 * fzero(@ColumnBuckling,5,opt,P,Sy,E,Leff,Fs,t)

function Pcr = ColumnBuckling(r,P,Sy,E,Leff,Fs,t)
Sr = 2 * Leff/sqrt(r^2 + (r - t)^2);
if Sr > pi * sqrt(2 * E/Sy)
  Pcr = P - (pi^3 * (r^2 - (r - t)^2) * E/Sr^2)/Fs;
else
  Pcr = P - pi * ((Sy - ((Sy * Sr)/(2 * pi))^2/E) * (r^2 - (r - t)^2))/Fs;
end
```

程序执行后得到 $d=8.9392$ in。

例 8.17　确定受偏心载荷作用的圆柱直径

再考虑前一个例子，圆柱偏心载荷的偏移量 $e=0.6$ in。对于这种情况，要创建两个函数：一个估计给定 r 时的 P_{cr}，另一个确定 $P=P_{cr}/F_s$ 时的 r。第一个函数是 **SetColumnBuckling**，第二个函数是 **EccenColumnBuckling**。程序如下：

```
funciotn ColumnBucklingDia
opt = optimset('display','off');
Leff = 2.1 * 12 * 13; P = 150000; Sy = 55000;
E = 3e + 7; Fs = 3, t = 0.75; ecc = 0.6;
d = 2 * fzero(@EccenColumnBuckling,6,opt,P,Sy,E,Leff,Fs,t,ecc)

function Pest = SecColumnBuckling(Pcr,r,t,Sy,E,Leff,Ecc)
a = (r^2 - (r - t)^2) * pi;
```

```
k = 0.5 * sprt(r^2 + (r - t)^2);
Pest = Pcr - a * Sy/(1 + (ecc * r/k^2) * sec(Leff/k * sqrt(Pcr/4/E/a)));

function Pcr = EccenColumnBucking(r,P,Sy,E,Leff,Fs,t,ecc)
options = optimset('display','off');
Pcr = P - fzero(@SecColumnBuckling,50000,options,r,t,Sy,E,Leff,ecc)/Fs;
```

执行程序后得到 $d = 9.6616$ in。

8.2.3 受交变载荷作用的轴

考虑一个实心圆钢轴分别受到一个完全相反变化的转矩 T_a 和弯曲力矩 M_a 的作用，平均转矩为 T_m 和平均弯曲力矩为 M_m；并且没有轴向载荷，其直径可以用下式估计[①]：

$$d = \sqrt[3]{\frac{32F_s}{\pi}} \sqrt[3]{\sqrt{(k_f M_a)^2 + 0.75(k_{fs} T_a)^2} / S_f + \sqrt{(k_{fm} M_m)^2 + 0.75(k_{fsm} T_m)^2} / S_{ut}} \quad (8.13)$$

其中，F_s 是安全系数，S_{ut}是材料的极限强度，S_f 是修正的疲劳强度，k_a 是变化应力的集中系数。当 M_a/M_m 和 T_a/T_m 为常数时，上式成立。很多值都是 d 的函数，这些函数定义如下：

修正疲劳强度 S_f

$$S_f = C_{size} C_{surf} C_{rel} S_{fu}$$

其中，对于圆形轴：

$C_{size} = 1$　　　　$d < 0.3$ in

$C_{size} = 0.869 d^{-0.097}$　　　　$0.3 \leqslant d \leqslant 10$ in

$C_{size} = 0.6$　　　　$d > 10$ in

对于机械表面 $C_{surf} = 2.7(S_{ut})^{-0.625}$且可靠性为 99% 时，$C_{rel} = 0.814$。在缺少相应数据的情况下，采用不精确的疲劳强度 S_{fu}和下面的粗糙度近似值应用于钢材：当 $S_{ut} < 200\ 000$ psi 时，$S_{fu} = 0.5S_{ut}$。基于这些假设，得到：

$$S_f = 1.0989 C_{size} S_{ut}^{0.735}$$

应力集中系数 k_a

应力集中疲劳系数由下式获得：

$$k_f = 1 + q(k_t - 1) \quad (8.14)$$

其中，k_t 是理论上的静态应力集中系数，q 是槽口灵敏度，是纽博尔常数 a 的函数(参见例 5.1)：

$$q = \left(1 + \sqrt{a/r}\right)^{-1}$$

其中，r 是槽口半径。

k_t 是载荷和几何形状的函数。研究轴直径突然从 D 变为 d 的情况，如图 8.17 所示。弯

① 参见 R.L.Norton，出处同上，第 575 页。

曲疲劳应力集中系数的值由下式估计[1]：

$$k_t = 1 + \left[A t_r^{-k} + B\left[\frac{1+a_r}{a_r^{3/2}}\right]^l + \frac{C a_r}{(a_r + t_r) t_r^m} \right]^{-1/2} \tag{8.15}$$

其中，

$$t_r = t / r_{\text{fillet}}$$

$$a_r = d/(2r_{\text{fillet}})$$

$$r_{\text{fillet}} < (D-d)/2$$

在不同弯曲和转矩条件下，A, B, C, k, l 和 m 的值由表 8.2 给定。

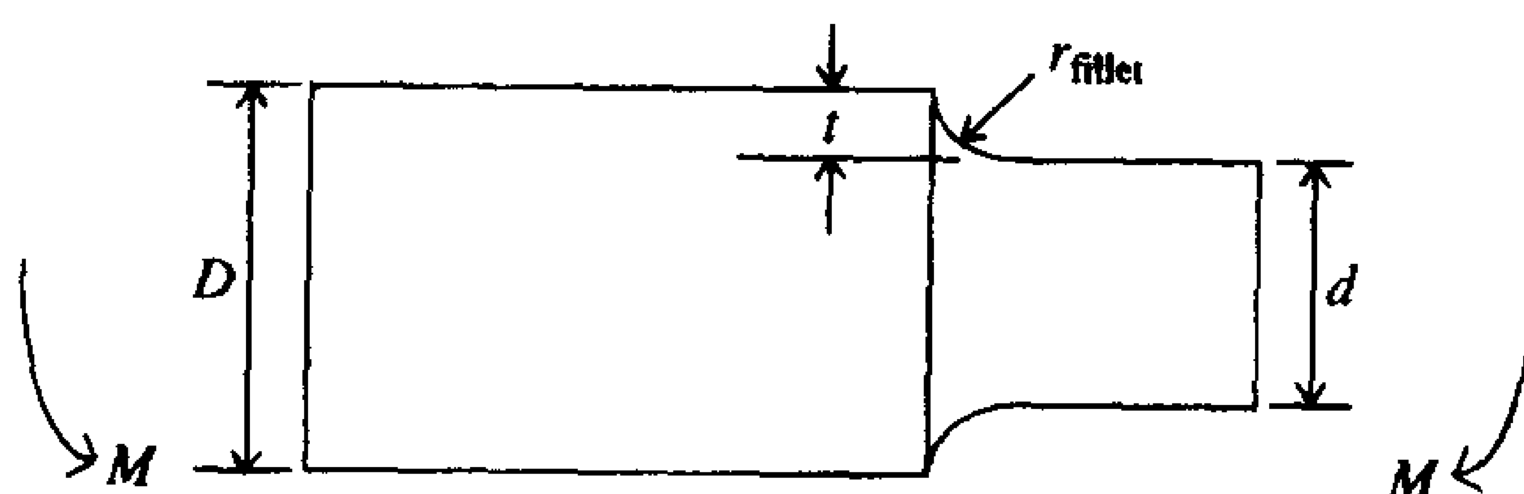


图 8.17　确定应力集中系数的几何尺寸和载荷

表 8.2　对应理论应力集中系数 k_t 常数的数值

常数	弯曲	转矩
A	0.40	0.40
B	6.00	25.0
C	0.80	0.20
k	0.40	0.45
l	2.75	2.25
m	1.50	2.00

k_{fm}的值由下式决定：

$k_f |\sigma_{max}| < S_y$

$$k_{fm} = k_f$$

$k_f |\sigma_{max}| > S_y$

$$k_{fm} = (S_y - k_f \sigma_a)/|\sigma_m|$$

$k_f |\sigma_{max} - \sigma_{max}| > 2S_y$

$$k_{fm} = 0$$

其中，k_f 的值由关于弯曲的式(8.14)和式(8.15)确定，上面的方程中给出的应力由下式决定：

$$\sigma_{max} = \frac{M_{max} r_{shaft}}{I} \qquad \sigma_{min} = \frac{M_{min} r_{shaft}}{I}$$

$$\sigma_a = \frac{M_a r_{shaft}}{I} \qquad \sigma_m = \frac{M_m r_{shaft}}{I}$$

其中，$r_{shaft} = d/2$。圆形截面轴的惯性力矩 $I = \pi d^4/64$，M_{max}是最大弯曲力矩，M_{min}是最小弯曲力矩，并且，

$$M_a = (M_{max} - M_{min})/2$$

---

① 参见 W. Beitz 和 K. H. Kuttner，编辑，*Handbook of Mechanical Engineering*，Springer-Verlag，New York，NY，1994 年，第 D78 页。

$$M_m = (M_{max} + M_{min})/2$$

k_{fsm}的值由下式决定：

$k_{fs}|\tau_{max}| < S_s$

$$k_{fsm} = k_{fs}$$

$k_{fs}|\tau_{max}| > S_s$

$$k_{fsm} = (S_s - k_{fs}\tau_a)/|\tau_m|$$

$k_{fs}|\tau_{max} - \tau_{max}| > 2S_s$

$$k_{fsm} = 0$$

其中，k_{fs}的值由转矩式(8.14)和式(8.15)确定，数值 S_s 是剪应力屈服强度，可由下式近似估计：

$$S_s \approx 0.58S_y$$

上面公式里的应力由下式给出：

$$\tau_{max} = \frac{T_{max} r_{shaft}}{J} \qquad \tau_{min} = \frac{T_{min} r_{shaft}}{J}$$

$$\tau_a = \frac{T_a r_{shaft}}{J} \qquad \tau_m = \frac{T_m r_{shaft}}{J}$$

其中，$J = \pi d^4/32$ 是圆轴截面的惯性矩，T_{max}是最大转矩，T_{min}是最小转矩，并且，

$$T_a = (T_{max} - T_{min})/2$$

$$T_m = (T_{max} + T_{min})/2$$

通过以下的例子可说明这些关系。

例 8.18　受交变载荷作用的轴的直径

一个机械钢圆轴受到的最大弯曲力矩和最小弯曲力矩分别为 4000 lb-in 和 1000 lb-in，并且受到的最大和最小转矩分别为 1600 lb-in 和 250 lb-in。另外，安全系数是 2.5，材料的屈服强度是 40 000 psi，极限强度为 70 000 psi，槽口半径是 0.03 in。D 比 d 大 15%，圆角半径是 d 的 10%，可靠性为 99% 的条件下轴心直径对于这个圆轴 $I = \pi d^4/64$，$J = \pi d^4/32$。

因为在式(8.13)中很多数值是轴直径的函数，所以，根据 d 的复化创建一个主函数 **FatigueDiameter**和四个子函数，使用 fzero 来估算 d。如果 $r_{fillet} = r_{fillet}/d$ 和 $Dod = D/d$，那么这 5 个函数为：

函数 **FatigueDiameter** 计算轴的直径 d

函数 **neuber** 计算 q(参见例 5.1，注意极限应力必须平均分成 1000 份)

函数 **StressConcenB** 计算 k_f 和 k_{fs}

函数 **StressConcenM** 计算 k_{fm}和 k_{fsm}

函数 **SsubF**，确定正确的疲劳应力 S_f

主函数和子函数为：

```
function diam = FatigueDiameter(d, rfilletd, Dod)
global Mmax Mmin Tmax Tmin Sy Su rnotch fs
Sf = SsubF(d);
D = Dod * d;
```

```
rfillet = rfilletd * d;
[kf,kfs] = StressConcenB(d,rfillet,D);
[kfm,kfsm] = StressConcenM(d,kf,kfs);
p1 = sqrt((kf * (Mmax-Mmin)/2)^2 + 0.75 * (kfs * (Tmax-Tmin)/2)^2)/Sf;
p2 = sqrt((kfm * (Mmax + Mmin)/2)^2 + 0.75 * (kfsm * (Tmax + Tmin)/2)^2)/Su;
diam = d - (32/pi * fs * (p1 + p2))^(1/3);

function Sf = SsubF(d)
global Mmax Mmin Tmax Tmin Sy Su rnotch fs
ifd < 0.3;
  csize = 1;
elseif d > 10;
  csize = 0.6;
else
  csize = 0.869 * d^( - 0.097);
end
Sf = 1.0989 * csize * Su^(0.735);
function [kfm,kfsm] = StressConcenM(d,kf,kfs)
global Mmax Mmin Tmax Tmin Sy Su rnotch fs
roj = 16/pi/d^3;
roi = 2 * /roj;
Smax = Mmax * roi;
Smin = Mmin * roi;
if kf * abs(Smax) <= Sy
  kfm = kf;
else
  kfm = (Sy - kf * roi * (Mmax-Mmin)/2)/abs(roi * (Mmax + Mmin)/2);
end
if kf * abs(Smax-Smin) > 2 * Sy;
  kfm = 0;
end
Tmax = Tmax * roj;
Tmin = Tmin * roj;
if kf * abs(Tmax) <= 0.58 * Sy
  kfsm = kfs;
else
  kfsm = (.58 * Sy-kfs * roj * (Tmax-Tmin)/2)/abs(roj * (Tmax + Tmin)/2);
end
if kf * abs (Tmax-Tmin) > 1.16 * Sy;
  kfsm = 0;
end

function q = neuber
global Mmax Mmin Tmax Tmin Sy Su rnotch fs
ncs = [15,.13;70,.092;90,.072;110,.057;130,.046;150,.037;...
  170,.028;190,.020;210,.015;230,.010;250,.007];
q = 1/(1 + polyval(polyfit(ncs(:,1),ncs(:,2),4),Su/1000)/sqrt(rnotch));

function[kf,kfs] = StressConcenB(d,rfillet,D)
A = [0.4,0.4];B = [6,25];C = [0.8,0.2];
k = [0.4,0.45];1 = [2.75,2.25];m = [1.5,2];
tr = (D - d)/2/rfillet;
```

```
ar = d/rfillet/2;
for n = 1:2
  t1 = A(n)/(tr)^k(n);
  t2 = B(n) * ((1 + ar)/(ar^1.5))^l(n);
  t3 = C(n) * ar/(ar + tr)/tr^m(n);
  alpha(n) = 1 + 1/sqrt(t1 + t2 + t3);
end
kf = 1 + neuber * (alpha(1) - 1);
kfs = 1 + neuber * (alpha(2) - 1);
```

程序为:

```
global Mmax Mmin Tmax Tmin Sy Su rnotch fs
opt = optimset('display','off');
Mmax = 4000; Mmin = 1000;
Tmax = 1600; Tmin = 250;
Sy = 40000; Su = 70000;
rnotch = 0.03; fa = 2.5;
rfilletd = 0.1; Dod = 1.15;
d = fzero(@FatigueDiameter,3,opt,rfilletd,Dod)
```

执行程序得到:$d = 2.6297$ in。

8.3 直齿圆柱齿轮的应力

在一个齿轮齿上的弯曲应力取决于一个均布(不变)的切向传输载荷 F_t,该弯曲应力由下式给出:

$$\sigma_b = \frac{K_v K_H F_t}{mbJ_K} \text{ N/mm}^2$$

其中,b 是齿轮齿的表面宽度,m 是模量,即 J_K 是弯曲强度的几何系数。K_v 是动载系数,该动载系数与轮齿的质量和齿节的切向速度 v_t 有关,K_H 是载荷分布系数。

切向载荷可由下式求出:

$$F_t = \frac{9.549 \times 10^6 P}{nR_p} = \frac{1000T}{R_p} \text{ N}$$

其中,P 是功率,单位是 kW;T 是扭矩,单位是 Nm;n 是两个齿轮中较小的一个齿轮转速,单位是 rpm。R_p 是两个齿轮中较小的操作齿节距,单位是 mm。

动载系数由下式估计:

$$K_v = \left(\frac{A + \sqrt{200 v_t}}{A}\right)^B$$

其中,

$$v_t = \frac{2\pi R_p n}{60\,000} \quad \text{m/s}$$

$$A = 50 + 56(1 - B)$$

$$B = 0.25(12 - Q_v)^{2/3}$$

Q_v 是齿轮的质量系数，取 $5 \leqslant Q_v \leqslant 11$ 区间内的整数。上限用于精密齿轮，下限用于最低精密度的齿轮。工作节圆半径是 R_p，单位为 mm；n 是转速，单位为 rpm；对于每一个 Q_v 值，v_t 有一个建议值，由下式给出：

$$v_{t_{max}} = (A + Q_v - 3)^2/200 \text{ m/s}$$

设计安装在轴承间的刚性齿轮时，不考虑外部相对偏差的影响。载荷分布系数可由下式估算：

$$K_H = 1 + K_{Hpf} + K_{Hma}$$

其中，K_{Hpf}是一个小齿轮的比例系数，K_{Hma}是啮合补偿系数。小齿轮的比例系数由下式估算：

$b \leqslant 25$ mm：

$$K_{Hpf} = k_o - 0.025$$

$25 < b \leqslant 432$ mm：

$$K_{Hpf} = k_o - 0.0375 + 0.000\,492b$$

$432 < b \leqslant 1020$ mm：

$$K_{Hpf} = k_o - 0.1109 + 0.000\,815b - 0.353 \times 10^{-6} b^2$$

其中，$k_o \geqslant 0.05$ 时，

$$k_o = 0.05b/R_p$$

其中，$k_o < 0.05$ 时，

$$k_o = 0.05$$

数值 R_p 是两个轮中小轮的工作节圆半径。

啮合补偿系数由下式估算：

$$K_{Hma} = A + Bb + Cb^2$$

其中，经验数据 A，B 和 C 由表 8.3 给出。

表 8.3　常数 A，B 和 C⁺

分类号	齿轮类型	A	B	C
1	开式	0.247	0.657×10^{-3}	-1.186×10^{-7}
2	闭式	0.127	0.622×10^{-3}	-1.69×10^{-7}
3	精密闭式	0.0675	0.504×10^{-3}	-1.44×10^{-7}
4	超精密闭式	0.0380	0.402×10^{-3}	-1.27×10^{-7}

⁺引自 AGMA 标准 2101-C95。

确定几何系数是很繁琐的，然而，美国齿轮制造商联合会（AGMA，American Gear Manufacturers Association）提供了一种图形化方法。将要用于代替由 Colbourne[①] 提出的计算方法。AGMA用相同的分析过程确定直齿圆柱齿轮的几何系数 J_K，不同的仅是找出轮齿临界的截面圆角。

计算过程中需要大量的术语和表达式，在表 8.4 和表 8.5 中列出，并在图 8.18 和图 8.19 中加以说明。参考这些图表，可以用下式估计几何系数：

$$J_K = \frac{\cos\varphi}{mK_J \cos\gamma_w}$$

① 参考 J.R. Colbourne, *The Geometry of Involute Gears*, Springer-Verlag, New York, NY, 1987 年。

其中,

$$K_J = K_f\left[\frac{1.5(x_D - x)}{y^2} - \frac{\tan\gamma_w}{2y}\right]_{\max}$$

$$K_f = k_1 + \left(\frac{2y}{r_f}\right)^{k_2}\left(\frac{2y}{x_D - x}\right)^{k_3}$$

并且,

$$k_1 = 0.3054 - 0.004\,89\varphi_s - 0.000\,069\varphi_s^2$$
$$k_2 = 0.3620 - 0.012\,68\varphi_s + 0.000\,104\varphi_s^2$$
$$k_3 = 0.2934 + 0.006\,09\varphi_s + 0.000\,087\varphi_s^2$$

其中,φ_s 是下面的齿轮压力角的度数表达:可以是 14.5°,20°或 25°。中括号$[\cdots]_{\max}$内的量表示 K_J 是最大值,由变量 x, y 确定。改变下式中的 u_r 可得到 x 和 y 的值,表 8.5 给出 u_r 的范围,如下:

$$u_{r\min} \leqslant u_r \leqslant u_{r\max}$$

x 和 y 值由下式给出:

$$x = R\cos\theta_R$$
$$y = R\sin\theta_R$$

其中,

$$R = \sqrt{(N_1 m/2 + \xi)^2 + \eta^2}$$

$$\theta_R = \tan\left(\frac{\eta}{N_1 m/2 + \xi}\right) - \frac{u_r - m\pi/2}{N_1 m/2}$$

$$\xi = s\xi' \qquad \eta = s\eta'$$

$$s = 1 + \frac{r_{fT}}{\sqrt{\xi'^2 + \eta'^2}}$$

$$\xi' = e + x_r', \quad \eta' = u_r + y_r'$$

表 8.4 用于确定直齿圆柱齿轮几何系数 J_k 参数的定义

参量	符号/公式
模数	m
齿轮齿数	N
基本节距	$p_b = m\pi\cos(\varphi_s)$
圆弧节距	$p_s = m\pi$
直径节距	$p_d = 1/m$
标准节圆半径	$R_s = Np_s/2\pi = Nm/2$
基圆半径	$R_b = R_s\cos(\varphi_s) = Np_b/2\pi = (Nm/2)\cos(\varphi_s)$
齿顶圆半径(齿坯半径)	R_T
齿顶高	$a = R_T - R_s$
(完全深度齿的标准齿顶高)	$(a = m)$
齿轮廓线上的点半径	R

（续表）

参量	符号/公式
工作节圆半径	$R_{P_1}=C/(1+N_2/N_1),R_{P_2}=C/(1+N_1/N_2)$
齿根半径	R_{root}
中心距	$C=R_{P_1}+R_{P_2}$
齿轮压力角	φ_s
齿条压力角	$\varphi_r(=\varphi_s)$
工作压力角	φ
在 R_s 处的齿厚	t_s
在 R 处的齿厚(参见图 8.18)	$t_R=2R\theta_R$
齿根圆角半径	r_f
齿条齿顶高 *	a_r
齿条铣刀的顶圆半径 *	r_{rT}
参考线和切割节线的距离[参考线在齿厚与齿间宽度相等的位置上($m\pi/2$),参见图 8.19]	e
渐开线函数	$\text{inv}(\varphi)=\tan(\varphi)-\varphi$

* 选择 a_r 和 r_{rT} 的取值,使之近似于与 m 相等,h 在表 8.5 中定义。

表 8.5 计算 J_k 时使用的几何值⁺

$e=\dfrac{t_s-m\pi/2}{2\tan\varphi_s}$	$m_c=\left[\sqrt{R_{T_1}^2-R_{b_1}^2}-\sqrt{R_{T_2}^2-R_{b_2}^2}-(R_{b_1}+R_{b_2})\tan\varphi\right]/p_b$
$r_f=r_{rT}+\dfrac{(a_r-e-r_{rT})^2}{mN_1/2+a_r-e-r_{rT}}$	$R_w^2=R_b^2+\left[\sqrt{R_T^2-R_b^2}-(m_c-1)p_b\right]^2$
$R_{bj}=N_j\pi\cos\varphi_s\quad j=1,2$	$\varphi_w=\cos(R_b/R_w)$
$\varphi=\cos\left(\dfrac{R_{b_1}+R_{b_2}}{C}\right)$	$\theta_w=\dfrac{t_s}{Nm}+\text{inv}\varphi_s-\text{inv}\varphi_w$
$x'_r=-a_r+r_{rT}$	$\gamma_w=\varphi_w-\theta_w$
$y'_r=m\pi/4+h\tan\varphi_s+r_{rT}\cos\varphi_s$	$x_D=R_w\cos\theta_w-R_w\sin\theta_w\tan\gamma_w$
$h=a_r-r_{rT}+r_{rT}\sin\varphi_s$	$u_{r\min}=\dfrac{e+x'_r}{\tan\varphi_s}-y'_r$
	$u_{r\max}=-y'_r$

⁺ 下标“1”表示要确定 J_k 的齿轮,下标“2”表示啮合齿轮。不带下标表示正计算 J_k 的齿轮。

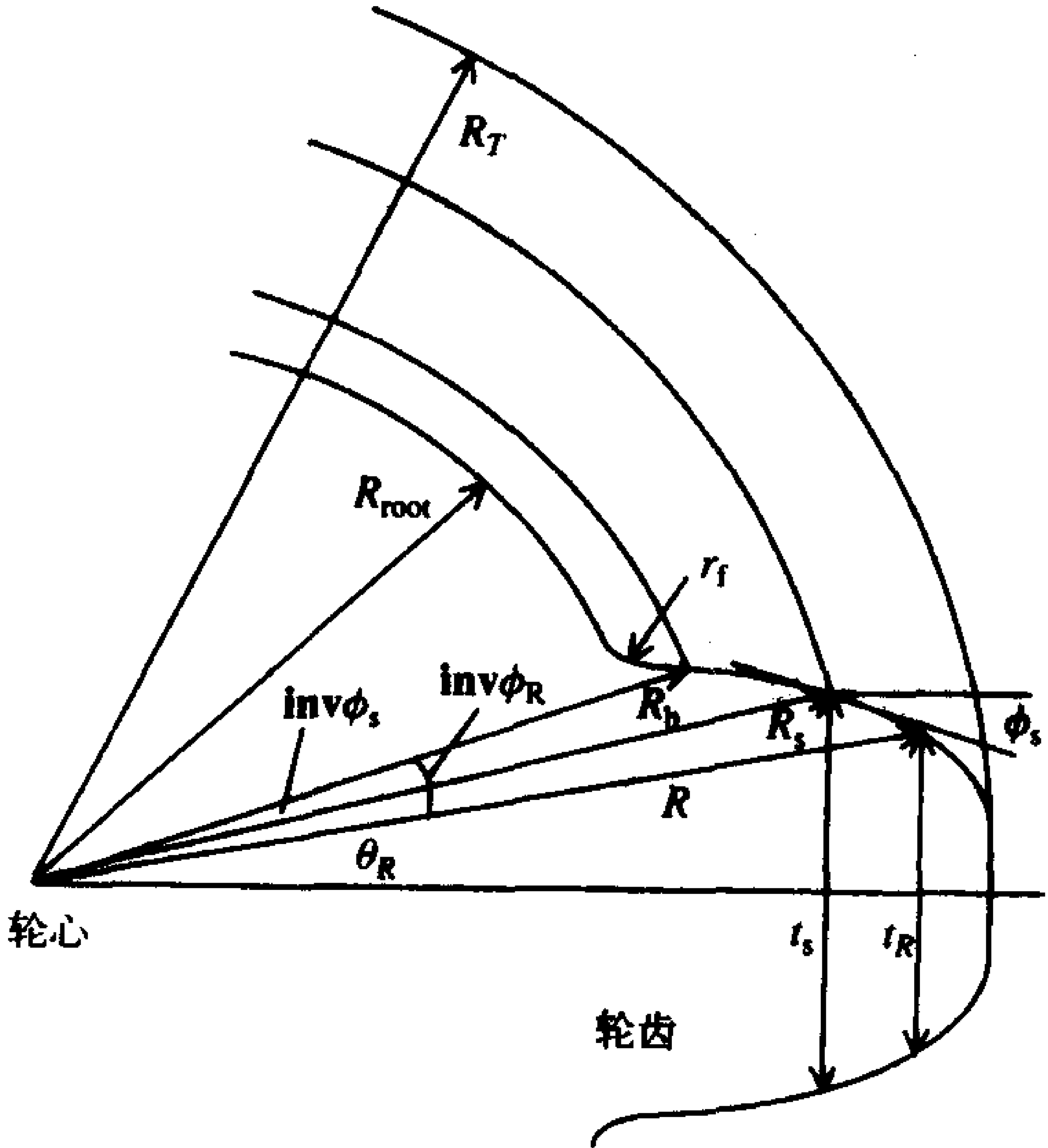


图 8.18 关于齿轮的术语

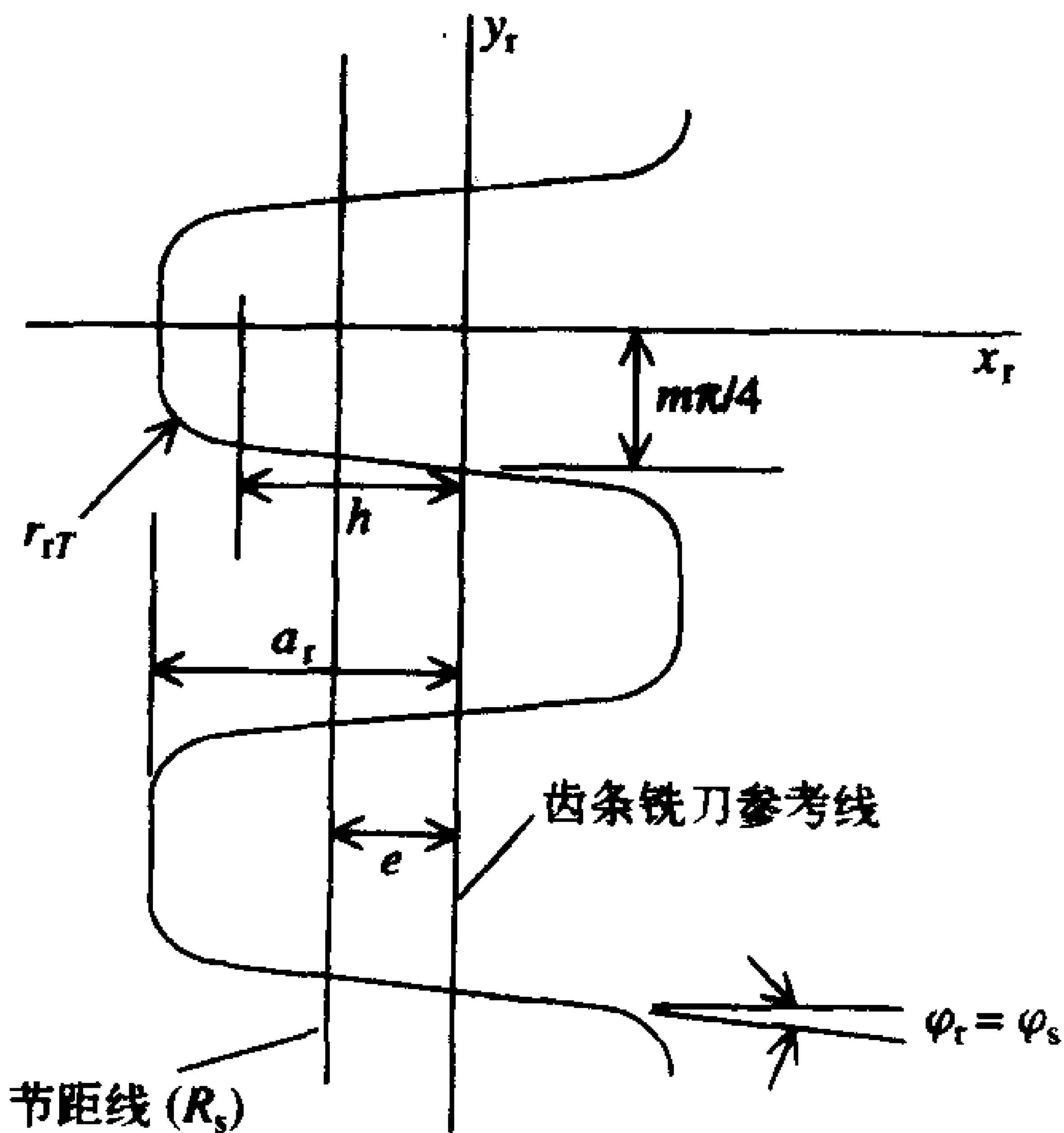


图 8.19 关于轮齿齿条铣刀的术语

一旦应力 σ_b 计算出来,许用应力 σ_F 便得以确定,并验证是否比由 F_t 引起的应力大,即 $\sigma_b \leqslant \sigma_F$。其中,许用应力是:

$$\sigma_F = \frac{\sigma_{FP} Y_N}{F_S Y_Z}$$

上式适用于油或齿轮温度低于 120℃的情况,数值 F_S 是安全系数,Y_N 是屈服应力循环系数,Y_Z 是可靠系数。可靠系数有如下值:$Y_Z = 1$ 表示失效率小于 1%,$Y_Z = 1.25$ 表示失效率小于 0.1%,$Y_Z = 1.5$ 表示失效率小于 0.01%。

如果 n_L 是单向齿载荷循环次数,那么在 99% 的可靠性内,钢齿轮的弯曲应力循环载荷系数可由下式估计:

$\boldsymbol{n_L \leqslant 3 \times 10^3}$:

$$Y_N = f(B_H)$$

其中,

$$f(B_H) = -9.2592 \times 10^{-6}(B_H - 160)^2 + 0.009\,722(B_H - 160) + 1.6 \qquad 160 \leqslant B_H \leqslant 400$$

而 B_H 是布里涅耳(Brinell)硬度值。

$\boldsymbol{3 \times 10^3 \leqslant n_L \leqslant 3 \times 10^6}$:

$$Y_N = D n_L^E$$

其中,

$$D = f(B_H) 10^{0.8628 C_1}$$
$$E = -0.2876 C_1$$
$$C_1 = \log_{10}[f(B_H)] - 0.0169$$

$\boldsymbol{n_L > 3 \times 10^6}$:

$$Y_N = 1.638 n_L^{-0.0323}$$

式中,B_H 是独立的。

循环次数由下式获得:

$$n_L = 60 n L$$

其中,L 是齿轮的寿命,单位按小时计;n 是齿轮的转速,单位是 rpm;并且假设节数是一。

对一级硬钢齿轮的许用弯曲应力数值是:

$$\sigma_{FP} = 0.533 B_H + 88.3 \text{ N/mm}^2$$

下面是使用这些关系的实例。

例 8.19　轮齿的弯曲强度

考虑以下几对齿轮以及制作这些齿轮的齿条铣刀的几何参数。

$m = 10$ mm	$a_r = 12.5$ mm	$R_{T_1} = 153.9$ mm
$\varphi_s = 20°$	$r_{rT} = 3.8$ mm	$R_{T_2} = 391.1$ mm
$N_1 = 28$	$b = 45$ mm	$n_1 = 1800$ r/m
$N_2 = 75$	$C = 525$ mm	$B_H = 260$
$T = 2500$ Nm	$t_s = 18.51$ mm	

注意,$h=10=m$,在允许的范围之内。假设安全系数是 1.2,齿轮将使用 4000 小时,期望的失效性低于 1%,这意味着 $Y_Z=1.0$。而且,假设 $Q_v=8$,并将齿轮安装在表 8.3 中类型 2 指定的密闭装置内。

创建主函数 *GearStress* 用来计算正齿轮的应力。创建 8 个子函数用来计算所需要的中间参数。子函数如下:

GearKofV——计算 K_v
GearKofH——计算 K_H
involute——计算角的渐开
GearParameters——计算表 8.5 中的数值
GearKofF——计算 K_f
GearKofJ——计算 K_J
GearJofK——计算 J_K
GearKofN——计算 Y_N

主函数及其子函数

```
function[sigmab,sigmaf] = GearStress
global m phis ar rrT ts C N1 N2 rT1 rT2 torque BH fs YZ b n Qv typeg L Rp Ft
JK = GrarJofK;
Kv = GearKofV;
KH = GearKofH;
sigmab = Kv * KH * Ft/m/b/JK;
sigmafp = 0.533 * BH + 88.3;
YN = GearYofN;
sigmaf = sigmafp * YN/fs/YZ;

function Kv = GearKofV
global Rp n Qv
vt = 2 * pi * Rp * n/60000;
B = 0.25 * (12 - Qv)^(2/3);
A = 50 + 56 * (1 - B);
vtmax = (A + Qv - 3)^2/200;
if vt > vtmax
  error('Maximum tangential velocity exceeded for given Qv')
end
Kv = ((A + sqrt(200 * vt))/A)^B;

function Kh = GearKofH
global b typeg Rp
class = [0.247 0.127 0.0675 0.0380;...
        0.657e-3 0.622e-3 0.504e-3 0.402e-3,...
        -1.186e-7 -1.69e-7 -1.44e-7 -1.27e-7];
Khma = class(1,typeg) + class(2,rypeg) * b + class(3,typeg) * b^2;
ko = 0.05 * b/Rp;
if ko < 0.05
  ko = 0.05;
end
if b <= 25
  Khpf = ko - 0.025;
  elseif b <= 432
```

```
        Khpf = ko - 0.0375 + 0.000492 * b;
    else
        Khpf = ko - 0.1109 + 0.000815 * b - 0.353e - 6 * b^2;
end
Kh = 1 + Khpf + Khma;

function GearParameters
global m phis ar rrT ts CN1 N2 rT1 rT2 t
global urmin urmax e xD rf xrp yrp gammaw phi
xrp = - ar + rrT;
h = ar - rrT + rrT * sin(phis);
yrp = - pi * m/4 + h * tan(phis) + rrT * cos(phis);
e = (ts - pi * m/2)/tan(phis)/2;
urmin = (e + xrp)/tan(phis) - yrp;
urmax = - yrp;
Pb = m * pi * cos(phis);
Rb1 = N1 * m/2 * cos(phis);
Rb2 = N1 * m/2 * cos(phis);
phi = acos((Pb1 + Pb2)/C);
mc = (sqrt(rT1^2 - Rb1^2) + sqrt(rT2^2 - Rb2^2) - (Rb1 + Rb2) * tan(phi)/Pb;
Rw = sqrt(Rb1^2 + (sqrt(rT1^2 - Rb1^2) - (mc-1) * Pb)^2);
phiw = acos(Rb1/Rw);
thetaw = ts/mN1 + involute(phis) - involute(phiw);
gammaw = phiw - thetaw;
xD = Rw * cos(thetaw) - Rw * sin(thetaw) * tan(gammaw);
rf = rrT + (ar - e - rrT)^2/(N1 * m/2 + ar - e - rrT);

function JK = GearJofK
global m phis ar rrT ts C N1 N2 rT1 rT2 t
global urmin urmax e xD rf xrp yrp gammaw phi
GearParameters
options = optimset('display','off');
ur = fminbad(@GearKofJ, urmin, urmax, options);
JK = - cos(phi)/GearKofJ(ur);

function KJ = GearKofJ(ur)
global m phis ar rrT ts C N1 N2 rT1 rT2 t
global urmin urmax e xD rf xrp yrp gammaw phi
xip = e + xrp;
etap = ur + yrp;
s = 1 + rrT/sqrt(xip^2 + etap^2);
xi = s * xip;
eta = s * etap;
thetaR = atan(eta/(N1 * m/2 + xi) - (ur - pi * m/2)/(N1 * m/2);
R = sqrt((N1 * m/2 + xi)^2 + eta^2);
x = R * cos(thetaR);
y = R * sin(thetaR);
Kf = GearKofF(x,y);
KJ = - m * cos(gammaw) * Kf * (1.5 * (xD - x)/y^2 - 0.5 * tan(gammaw)/y);

function Kf = GearKofF(x,y)
global phis rf xD
d = phis * 180/pi;
```

```
k1 = 0.3054 - 0.00489* d - 0.000069* d^2;
k2 = 0.362 - 0.01268* d + 0.000104* d^2;
k3 = 0.2934 + 0.00609 * d + 0.0000087 * d^2;
KF = k1 + (2* y/rf)^k2* (2* y/(xD - x))^k3;

function inv = involute(angle)
inv = tan(angle) - angle;

function YN = GearYofN
global n L BH
nL = 60* n* L;
fBH = -9.2595e-6* (BH-160)^2 + 0.009722* (BH - 160) + 1.6;
if nL <= 1e3
  YN = fBH;
elseif nL <= 3e6
  D = 0.8628 * (log10(fBH) - 0.0169);
  E = -0.2876 * (log10(fBH) - 0.0169);
  YN = (fBH * 10^D) * nL.^E;
else
  YN = 1.683 * nL^(-0.0323);
end
```

函数 **GearJofK** 通过使用 fminbnd 来确定 K_J 的最大值。其中，K_J 是 u_r 的函数。为了让 fminbnd 正确运行，需要改变函数中数值 **GearJofK** 的符号。所以在 **GearJofK** 的最后一行中插入了一个负号，用来改变 J_K 的符号。

确定弯曲应力和许用应力的程序如下：

```
global m phis ar rrT ts C N1 N2 rT1 rT2 torque BH fs YZ b n Qv rypeg L Rp Ft
m = 10; phis = 20 * pi/180; ar = 12.5; rrT = 3.8; ts = 18.51;
C = 525; N1 = 28; N2 = 75; rT1 = 153.9; rT2 = 391.1; torque = 2400;
BH = 260; fs = 1.2; YZ = 1.0; b = 45; n = 1800; Qv = 8; typeg = 2; L = 4000;
Rp = C/(1 + N2/N1);
Ft = 1000 * torque/Rp;
[sigmab, sigmaf] = GearStress;
disp(['The bending stress is' num2str(sigmab,5)' N/mm^2'])
disp(['The permissible level is' num2str(sigmaf,5)' N/mm^2'])
```

程序执行后，在 MATLAB 命令窗口显示以下内容：

```
The bending stress is 149.64 N/mm^2
The permissible level is 167.41 N/mm^2
```

8.4 四连杆机构的运动学

8.4.1 连杆的位置和速度

研究如图 8.20 所示的机构。当角度 θ_1 和 θ_2 给定时，θ_3 和 θ_4 由下式给出：

$$
\begin{aligned}
L_2\cos\theta_2 + L_3\cos\theta_3 - L_4\cos\theta_4 - L_1\cos\theta_1 &= 0 \\
L_2\sin\theta_2 + L_3\sin\theta_3 - L_4\sin\theta_4 - L_1\sin\theta_1 &= 0
\end{aligned}
\tag{8.16}
$$

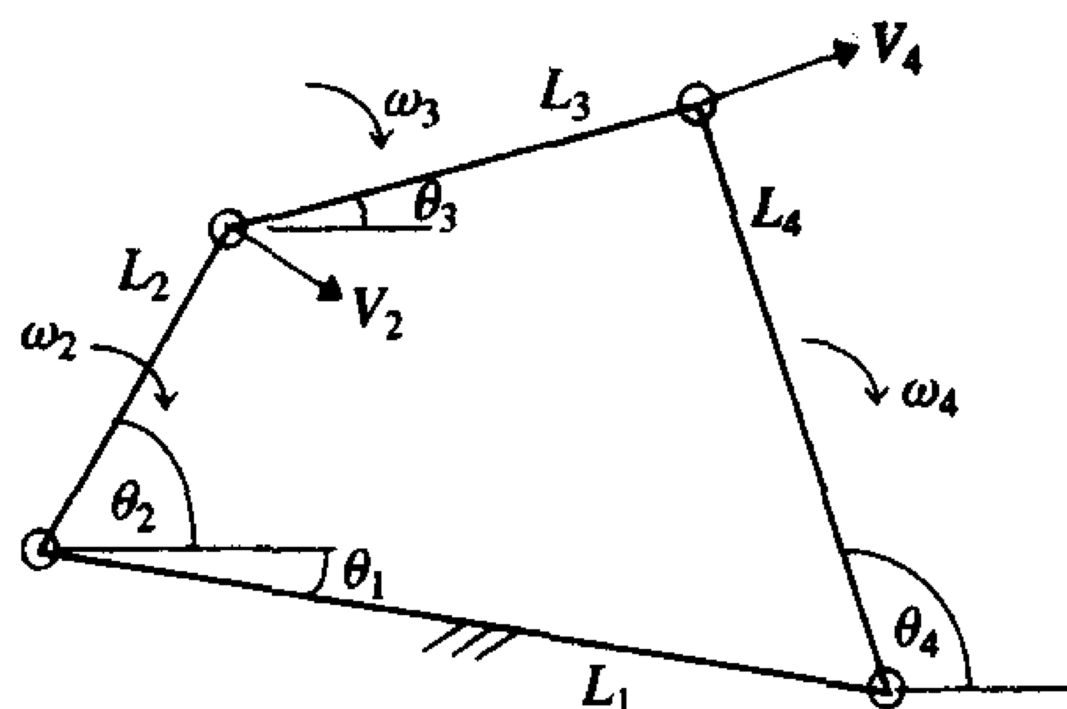


图 8.20 四连杆机构的术语

为方便起见,令 $\theta_1=0$。如果连杆 2 的角速度给定,则连杆 3 和连杆 4 的角速度分别是:

$$\omega_3=\frac{L_2\omega_2\sin(\theta_4-\theta_2)}{L_3\sin(\theta_3-\theta_4)}\qquad \omega_4=\frac{L_2\omega_2\sin(\theta_2-\theta_3)}{L_4\sin(\theta_4-\theta_3)}$$

其中,θ_3 和 θ_4 由式(8.16)得到。连杆 2 端部的线速度是 $V_2=L_2\omega_2$,连杆 4 端部的线速度是:

$$V_4=L_4\omega_4=V_2\frac{\sin(\theta_2-\theta_3)}{\sin(\theta_4-\theta_3)}$$

每一种速度在角速度 ω_j 的方向上分别垂直于杆 L_j。

连杆 3 和连杆 4 的角加速度是[①]:

$$\alpha_3=\frac{-L_2\alpha_2\sin(\theta_4-\theta_2)+L_2\omega_2^2\cos(\theta_4-\theta_2)+L_3\omega_3^2\cos(\theta_4-\theta_3)-L_4\omega_4^2}{L_3\sin(\theta_3-\theta_4)}$$

$$\alpha_4=\frac{L_2\alpha_2\sin(\theta_3-\theta_2)-L_2\omega_2^2\cos(\theta_3-\theta_2)+L_4\omega_4^2\cos(\theta_3-\theta_4)-L_3\omega_3^2}{L_4\sin(\theta_3-\theta_4)}$$

用代数求解法可以从式(8.16)直接获得 θ_3 和 θ_4 的表达式,但很烦琐。同样,也可以通过优化工具箱中的 fsolve 对这些方程数值求解,如下例所示。

例 8.20 四连杆机构的位置、速度和加速度的图形化表示

在 $L_1=0.8, L_2=2, L_3=2$ 和 $L_4=3$ 的情况下,画出连杆 3 的方向,在另一幅图中绘出速度比 V_4/V_2 和 $\omega_2=4$ rad/s 及 $\alpha_2=5$ rad/s^2 时的加速度 α_4。子函数 **FourBarPosition** 由 **fsolve** 用来确定 θ_3 和 θ_4。在这个函数中 $th(1)=\theta_3, th(2)=\theta_4, th1=\theta_1$ 和 $th2=\theta_2$。

显示连杆 3 的位置、连杆 4 的端部速度和加速度的程序如下:

```
function FourBarLinakge
L2 = .8;L3 = 2;L4 = 2;L1 = 3;th1 = 0;
th2 = [1/6:1/6:2] * pi;
th34 = zeros(length(th2),2);
opt = optimset('display','off');
for m = 1:length(th2)
  th34(m,:) = fsolve(@FourBarPosition,[5 5],opt,th2(m),th1,L2,L3,L4,L1);
end
```

① 参见 A.G.Erdman 和 G.N.Sandor, *Mechanism Design: Analysis and Synthesis*,第二版,Prentice Hall,Upper Saddle River,NJ,1991 年,第 231 页.

```
y = L2 * sin(th2) + L3 * sin(th34(:,1)');
x = L2 * cos(th2) + L3 * cos(th34(:,1)');
xx = [L2 * cos(th2)];
yy = [L2 * sin(th2)];
figure(1)
plot([x;xx],[y;yy],'k',[0 L1],[0 0],'k--^',x,y,'ko',xx,yy,'ks')
hold on
th = linspace(0,2 * pi,100);
plot(L2 * cos(th),L2 * sin(th),'k--')
xlabel('Horizontal position')
ylabel('Vertical position')
axis equal
th2 = [0:05:2] * pi;
th34 = zeros(length(th2),2);
for m = 1:length(th2)
  th34(m,:) = fsolve(@FourBarPosition,[5 5],opt,th2(m),th1,L2,L3,L4,L1);
end
figure(2)
y = sin(th2 - th34(:,1)')./sin(th34(:,2)'-th34(:,1)');
plot(180 * th2/pi,y)
v = axis;
v(2) = 360;
axis(v)
xlabel('\theta_2(degrees)')
ylabel('V_4/V_2')
w2 = 4;alph2 = 5;
w3 = (L2 * w2 * sin(th34(:,2)-th2'))./(L3 * sin(th34(:,1)-th34(:,2)));
w4 = (L2 * w2 * sin(th2'-th34(:,1)))./(L4 * sin(th34(:,2)-th34(:,1)));
s32 = th34(:,1)-th2';
s34 = th34(:,1)-th34(:,2);
alph4 = (L2 * alph2 * sin(s32)-L2 * w2^2 * cos(s32) + L4 * w4.^2. * cos(s34)...
       -L3 * w3.^2)./(L4 * sin(s34));
figure(3)
plot(180 * th2/pi,alph4)
v = axis;
v(2) = 360;
axis(v)
xlabel('\theta_2(degrees)')
ylabel('\alpha_4')

function t = FourBarPosition(th,th2,th1,L2,L3,L4,L1)
t = [L2 * cos(th2) + L3 * cos(th(1))-L4 * cos(th(2))-L1 * cos(th1);...
   L2 * sin(th2) + L3 * sin(th(1))-L4 * sin(th(2))-L1 * sin(th1)];
```

程序执行结果如图 8.21、图 8.22 和图 8.23 所示。

8.4.2 四连杆机构的综合

四连杆机构综合的目标之一是确定连杆的长度，使游动杆上选定的点通过三个规定的点。研究如图 8.24 所示的机构。目标是确定长度 Z_k 及其初始方向，以使点 P 通过点 P_1，P_2 和 P_3。假设点 P_1 在坐标系的原点处，使 $R_1 = 0$。

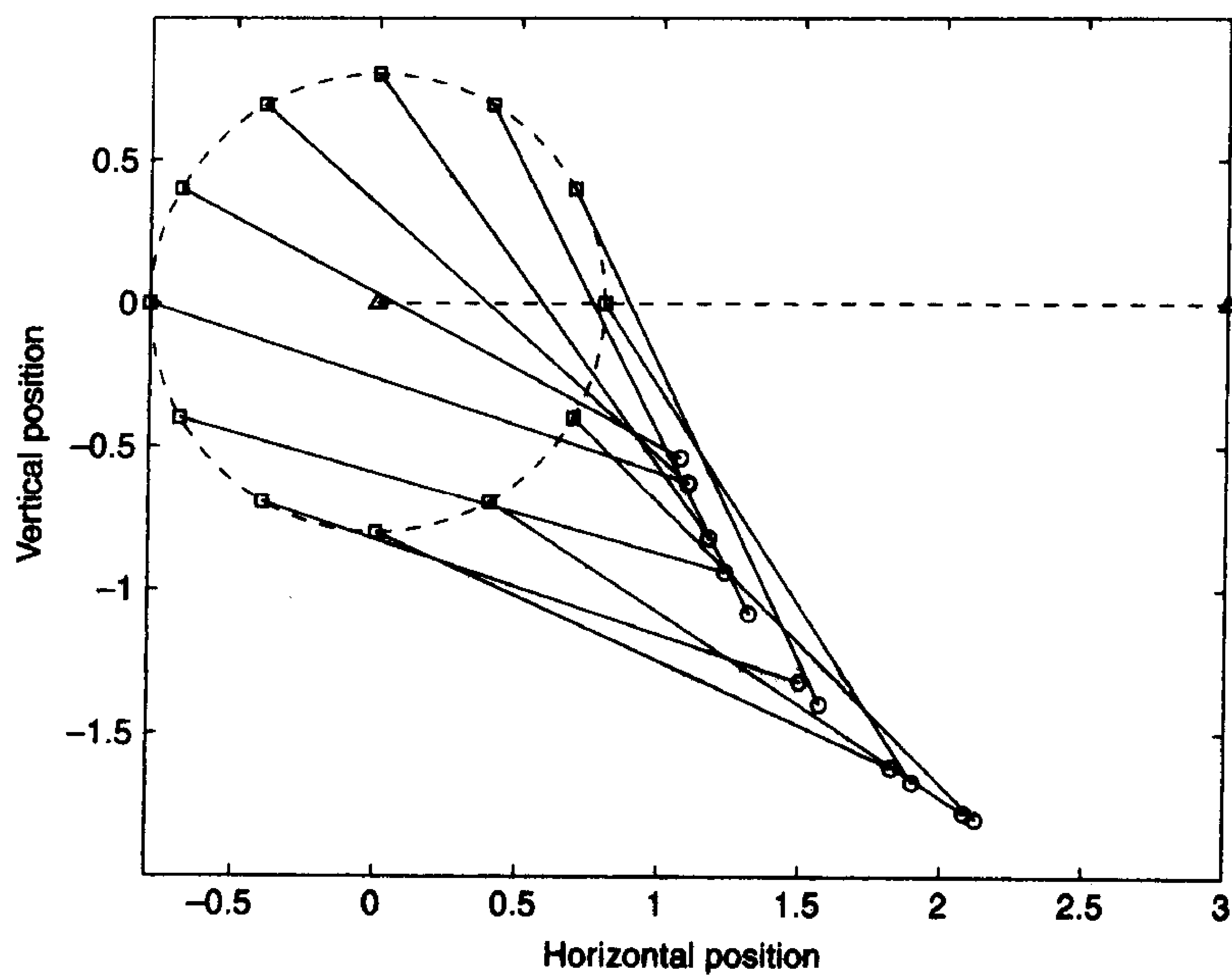


图 8.21　图 8.20 中连杆 3 的几个位置点

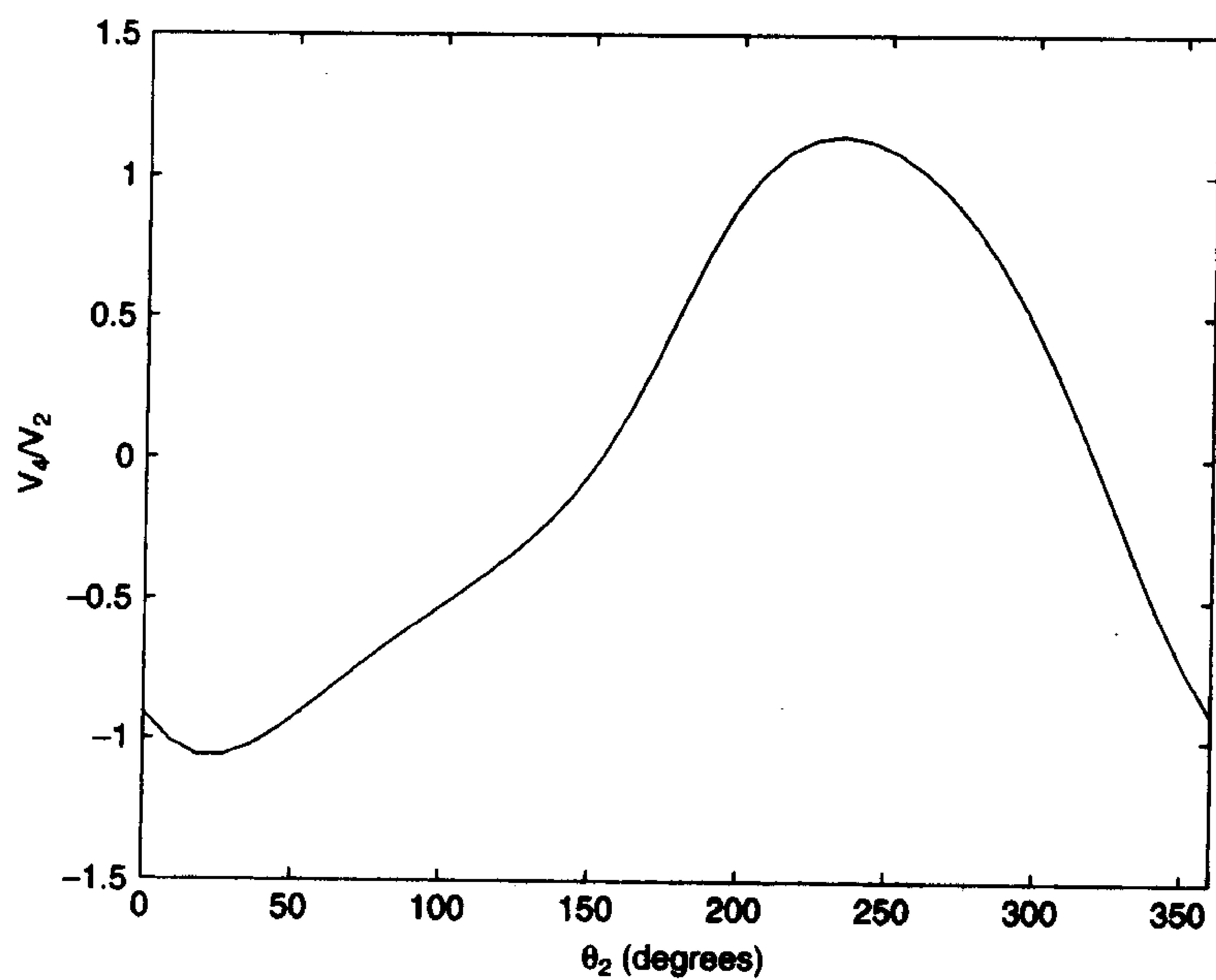


图 8.22　图 8.20 中连杆 4 的端部速度

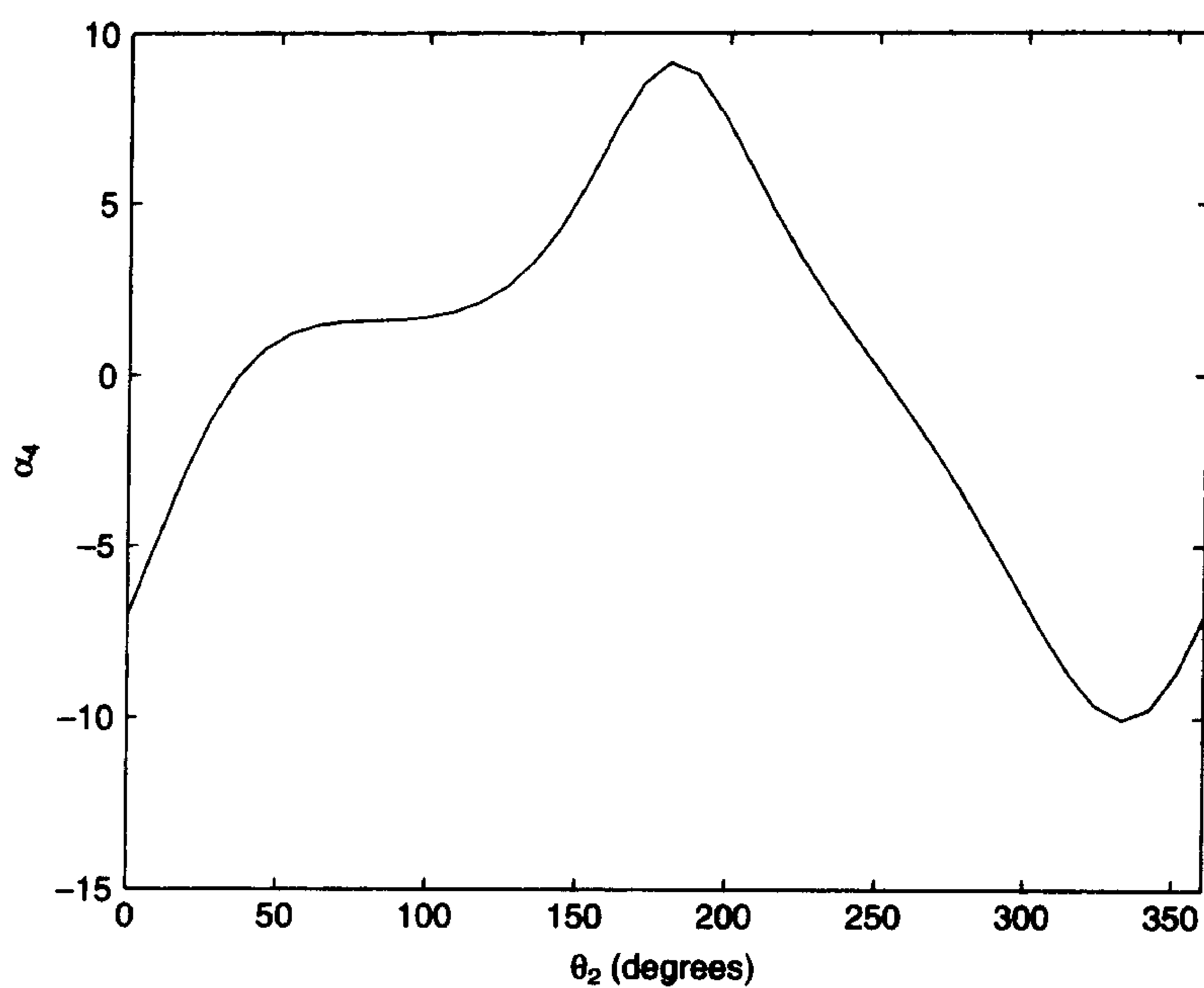


图 8.23　图 8.20 中连杆 4 的角加速度

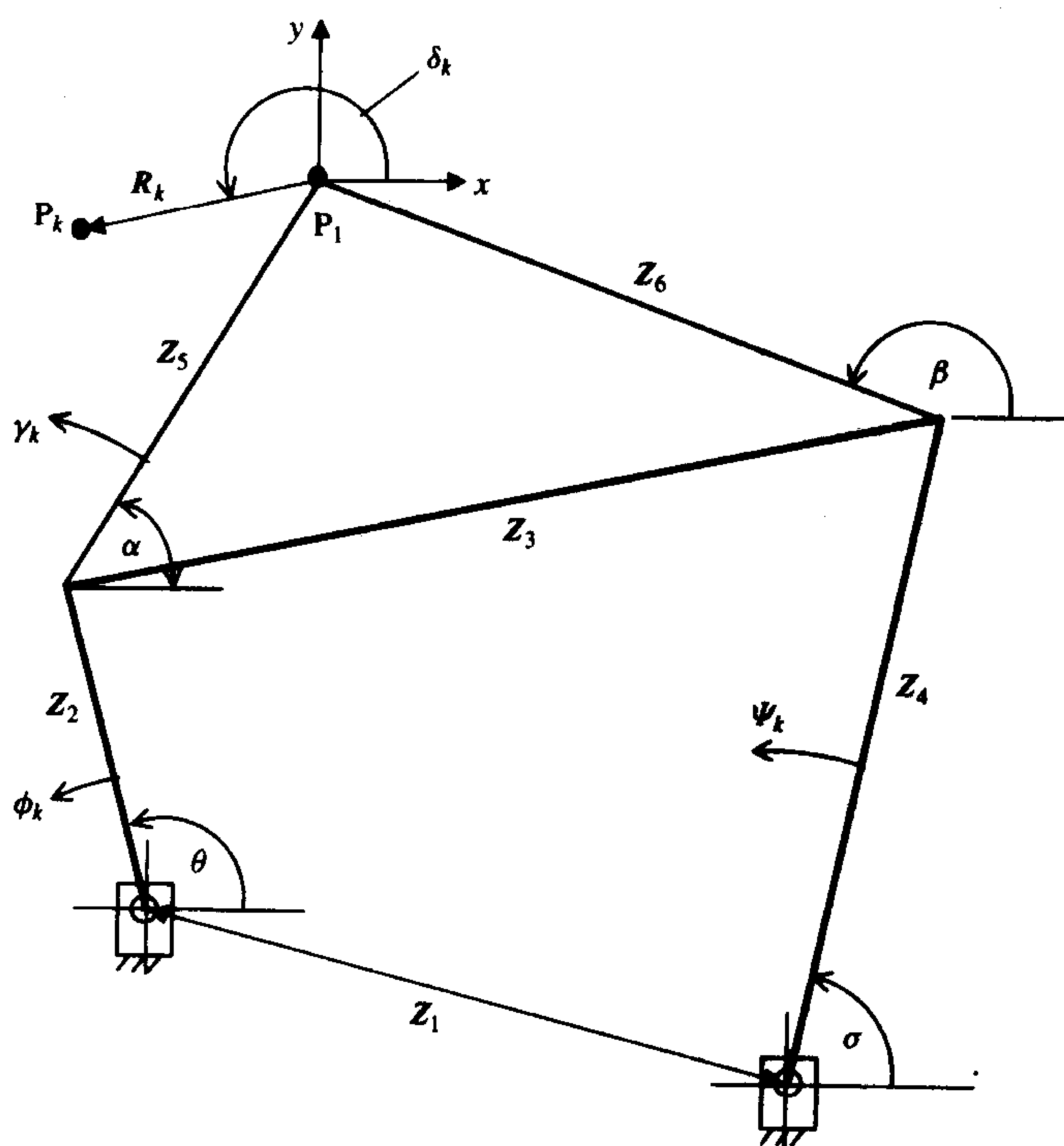


图 8.24　三点综合的向量术语

计算六个连杆长度的方程用二维向量给出,该向量可以很容易地表示为复数。六个向量的长度和方向可以由下式确定[1]:

$$Z_2 = \frac{1}{D}\begin{vmatrix} R_1 & d_{12} \\ R_2 & d_{22} \end{vmatrix} \qquad Z_5 = \frac{1}{D}\begin{vmatrix} d_{11} & R_1 \\ d_{21} & R_2 \end{vmatrix}$$

$$Z_4 = \frac{1}{E}\begin{vmatrix} R_1 & e_{12} \\ R_2 & e_{22} \end{vmatrix} \qquad Z_6 = \frac{1}{E}\begin{vmatrix} e_{11} & R_1 \\ e_{21} & R_2 \end{vmatrix}$$

$$Z_3 = Z_5 - Z_6 \qquad Z_1 = Z_2 + Z_3 - Z_4$$

$$D = \begin{vmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{vmatrix} = \begin{vmatrix} e^{j\phi_1} - 1 & e^{j\gamma_1} - 1 \\ e^{j\phi_2} - 1 & e^{j\gamma_2} - 1 \end{vmatrix}$$

$$E = \begin{vmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{vmatrix} = \begin{vmatrix} e^{j\Psi_1} - 1 & e^{j\gamma_1} - 1 \\ e^{j\Psi_2} - 1 & e^{j\gamma_2} - 1 \end{vmatrix}$$

其中,

$$R_k = |R_k| e^{j\delta_k}$$

比较图 8.20 和图 8.24 可以看出,$|Z_1| = L_1$,$|Z_2| = L_2$,$|Z_3| = L_3$,$|Z_4| = L_4$。

对 R_k, ϕ_k, γ_k 和 Ψ_k,$k = 1,2$ 的值进行假设。在这种情况下能确定 6 个连杆的长度和方向角 α, β, σ 和 θ。下面用例子来验证这个结论。

例 8.21　四连杆机构的综合

设图 8.24 系统中的数值如下所示:

$\phi_1 = 340°$	$\gamma_1 = -48°$	$\delta_1 = -31°$
$\phi_2 = 325°$	$\gamma_2 = 9°$	$\delta_2 = -15°$
$\Psi_1 = 31°$	$\|P_1\| = 2.7$	
$\Psi_2 = 81°$	$\|P_2\| = 3.9$	

程序如下:其中子函数 **FourBarSynth** 用来求成对齿轮 Z_2 和 Z_5,Z_4 和 Z_6 的值。

```
function FourBarLinkage2
c = pi/180;phi = [340.0 325.0] * c;gama = [ - 48 9.0] * c;
R = [2.7 * exp( - 31.0 * j * c),3.9 * exp( - 15.0 * j * c)];
z25 = FourBarSynth(phi,gama,R);
disp(['Z2 = 'num2str(abs(z25(1)))'theta = '...
          num2str(angle(z25(1))/c)'degrees'])
disp(['Z5 = 'num2str(abs(z25(2)))'alpha = '...
          num2str(angle(z25(2))/c)'degrees'])
psi = [31 80] * c;
z46 = FourBarSynth(psi,gama,R);
disp(['Z4 = 'num2str(abs(z46(1)))'sigma = '...
         num2str(angle(z46(1))/c)'degrees'])
disp(['Z6 = 'num2str(abs(z46(2)))'beta = '...
         num2str(angle(z46(2))/c)'degrees'])
Z3 = z25(2)-z46(2);
```

① 参见 A.G.Erdman 和 G.N.Sandor, 出处同上,第 530 ~ 532 页。

```
Z1 = z25(1) + Z3-z46(1);
disp('Z3 = 'num2str(abs(Z3))])
disp('Z1 = 'num2str(abs(Z1))])

function z = FourBarSynth(phi,gama,R)
coeff = [exp(j * phi(1)) - 1 exp(j * gama(1)) - 1;exp(j * phi(2)) - 1 exp(j * gama(2)) - 1];
D = det(coeff);
z1 = det([R(1)coeff(1,2);R(2)coeff(2,2)])/D;
z2 = det([coeff(1,1)R(1);coeff(2,1)R(2)])/D;
z = [z1 z2];
```

执行程序,在 MATLAB 命令窗口中显示以下信息:

```
Z2 = 6.7143   theta = 90.5276 degrees
Z5 = 1.2378   alpha = 24.8126 degrees
Z4 = 3.1762   sigma = - 143.832 degrees
Z6 = 1.2774   beta = 95.0336 degrees
Z3 = 1.447
Z1 = 8.6814
```

这些结果显示在图 8.25 中,用来生成图形的程序留待在习题 8.5 中完成。

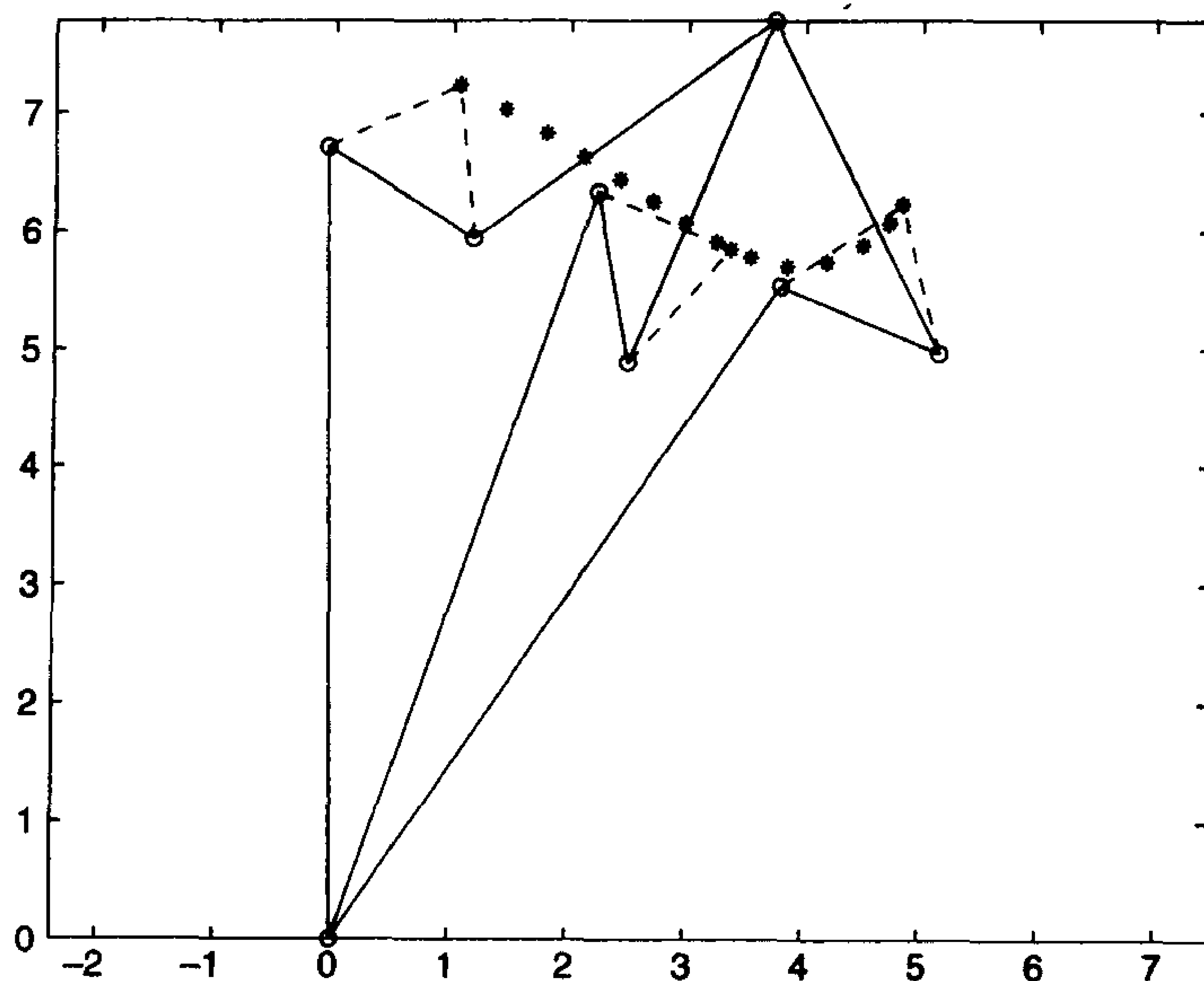


图 8.25　被综合的四连杆机构 P 点的轨迹

8.5 凸轮轮廓及其综合

8.5.1 凸轮位移

凸轮是将一种运动转化为另一种运动的装置。凸轮有一个向外弯曲的或是一个向内凹进的表面,并与从动件一起实现运动形式的转换。凸轮通常是旋转运动形式,旋转运动可以转化成振动、直线运动或是两者的结合。应该考虑旋转运动的情况下对于从动件的位移轮廓。有

两种类型的从动件,一种是表面盘形从动件,另一种是偏轴滚子从动件。综合的目的是确定凸轮轮廓和给定半径的剪床坐标,这些都是用来生成凸轮轮廓的。

定义一个凸轮基圆 r_b 作为最小的圆周半径,它与凸轮表面相切,而且和凸轮的转轴同心。从动件的运动方程如下:

$$L(\varphi) = r_b + s(\varphi)$$

对于旋转运动有:

$$s(\varphi) = h\left(\frac{\varphi}{\beta} - \frac{1}{2\pi}\sin(2\pi\varphi/\beta)\right) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$s(\varphi) = h - h\left(\frac{\varphi - \beta}{\beta} - \frac{1}{2\pi}\sin(2\pi(\varphi - \beta)/\beta)\right) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$s(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

上式是从动件的位移,h 是从动件的最大位移,并且 $0 \leqslant \beta \leqslant \pi$。

如果假设凸轮的旋转速度 $\omega = d\varphi/dt$ 是个常量,速度 ν、加速度 a 和瞬时加速度 j(加速度对时间求导)如果定义无量纲位移 $S = s/h$,无量纲速度 $V = v/(\omega h)$,无量纲加速度 $A = a/(h\omega^2)$,以及无量纲瞬时加速度 $J = j/(h\omega^3)$,就会有以下的关系式:

位移:

$$S(\varphi) = \frac{\varphi}{\beta} - \frac{1}{2\pi}\sin(2\pi\varphi/\beta) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$S(\varphi) = 1 - \left(\frac{\omega - \beta}{\beta} - \frac{1}{2\pi}\sin(2\pi(\varphi - \beta)/\beta)\right) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$S(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

速度:

$$V(\varphi) = \frac{1}{\beta}(1 - \cos(2\pi\varphi/\beta)) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$V(\varphi) = -\frac{1}{\beta}(1 - \cos(2\pi(\varphi - \beta)/\beta)) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$V(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

加速度:

$$A(\varphi) = \frac{2\pi}{\beta^2}\sin(2\pi\varphi/\beta) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$A(\varphi) = -\frac{2\pi}{\beta^2}\sin(2\pi(\varphi - \beta)/\beta) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$A(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

瞬时加速度:

$$J(\varphi) = \frac{4\pi^2}{\beta^3}\cos(2\pi\varphi/\beta) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$J(\varphi) = -\frac{4\pi^2}{\beta^3}\cos(2\pi(\varphi - \beta)/\beta) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$J(\varphi) = 0 \qquad 2\beta < \varphi < 2\pi$$

以下通过例子说明这些结论的使用。

例 8.22　摆线凸轮的位移、速度、加速度和瞬时加速度

计算当 $\beta = 60°$时，凸轮的无量纲位移、速度、加速度和瞬时加速度的值，并绘出图形。程序如下：

```
beta = 60 * pi/180;N = 40;
phi = linspace(0,beta,N);
phi2 = beta + phi;
ph = [phi,phi2] * 180/pi;
arg = 2 * pi * phi/beta;
arg2 = 2 * pi * (phi2-beta)/beta;
cam = zeros(4,2 * N)
lab = char('Displacement(S)','Velocity(V)','Acceleration(A)','Jerk(J)');
cam(1,:) = [phi/beta-sin(arg)/2/pi,1-(arg2-sin(arg2))/2/pi];
cam(2,:) = [(1-cos(arg))/beta,-(1-cos(arg2))/beta];
cam(3,:) = [2 * pi/beta^2 * sin(arg),-2 * pi/beta^2 * sin(arg2)];
cam(4,:) = [4 * pi^2/beta^3 * cos(arg),-4 * pi^2/beta^3 * cos(arg2)];
for k = 1:4
  subplot(2,2,k)
  plot(ph,cam(k,:),'k',[0 120],[0 0],'k--')
  xlabel('Cam angle(degrees)')
  ylabel(lab(k,:))
  g = axis;
  g(2) = 120;
  axis(g)
end
```

程序运行后的结果如图 8.26 所示。

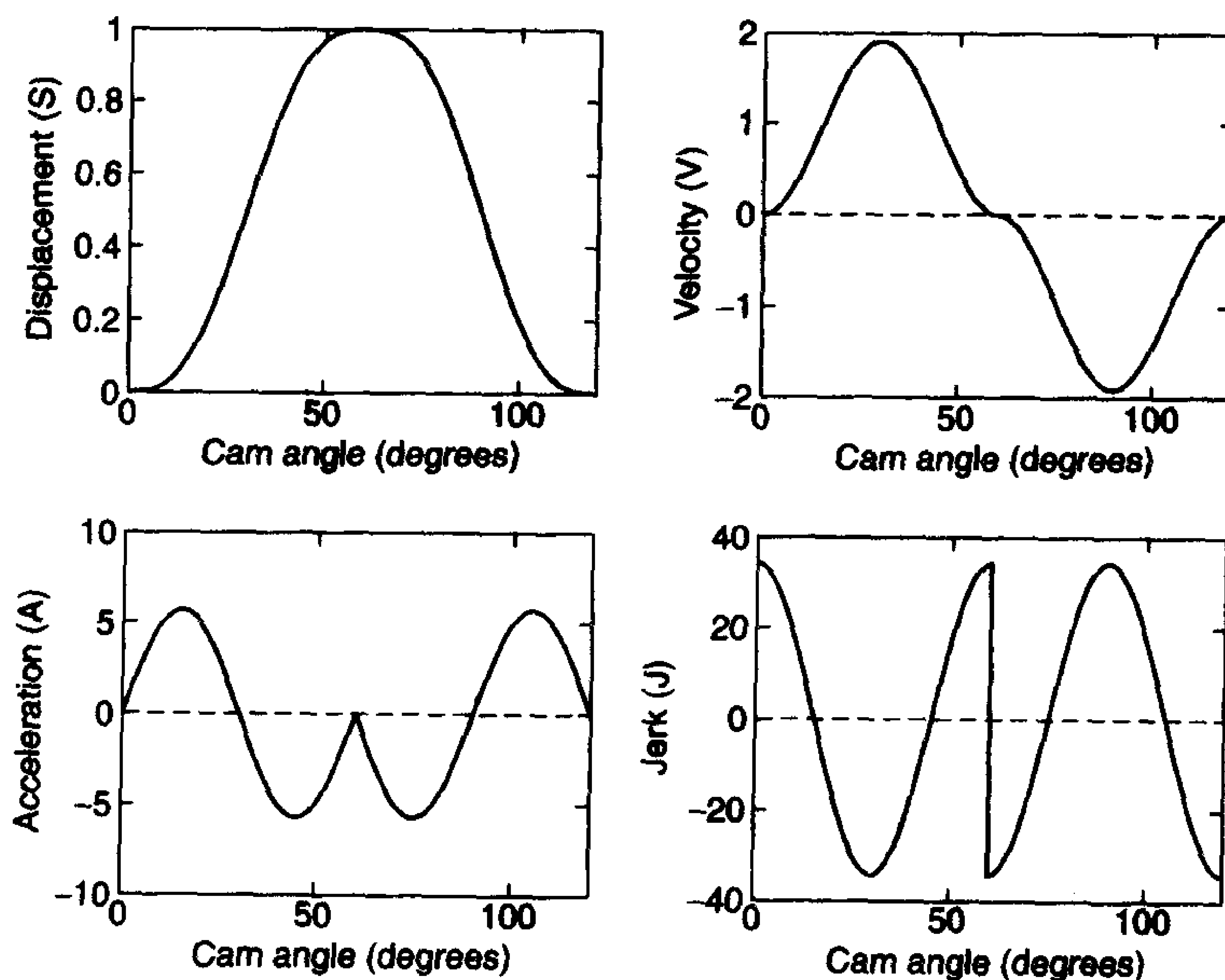


图 8.26　摆线凸轮轮廓的标准位移、速度、加速度和瞬时加速度

8.5.2　表面盘形从动件

参考图 8.27 得到以下关系[①]:在(x,y)坐标系中,凸轮轮廓的坐标为 R_x 和R_y,刀具的坐标为 C_x 和C_y:

$$R_x = R\cos(\theta+\varphi) \qquad R_y = R\sin(\theta+\varphi)$$
$$C_x = C\cos(\gamma+\varphi) \qquad C_y = C\sin(\gamma+\varphi)$$

其中,

$$R = \frac{L}{\cos\theta} \qquad \theta = \tan^{-1}\left(\frac{1}{L}\frac{dL}{d\varphi}\right)$$
$$C = \frac{L+r_c}{\cos\gamma} \qquad \gamma = \tan^{-1}\left(\frac{dL/d\varphi}{L+r_c}\right)$$

r_c 是刀具的半径,且 $dL/d\varphi = hV(\varphi)$。其中,$V(\varphi)$在 8.5.1 节中已经给出以下这个例子来说明这些结论。

图 8.27　带转换平面从动件的凸轮

例 8.23　表面盘形从动件的凸轮轮廓和刀具的坐标系

当 $\beta=60°$,$r_b=3.0$,$h=0.5$ 时,计算 R_x,R_y,C_x 和C_y,并绘出图形。在此使用两个函数,

① 参见 A.G.Erdman 和 G.N.Sandor,出处同上,第 385 ~ 387 页和 H.A.Rothbart, *Cams: Design, Dynamics, Accuracy*, John Wiley & Sons, New York, NY, 1956 年,第 122 ~ 124 页。

函数 **CamProfile** 用来计算一系列 φ 值对应的 $L(\varphi)$ 和 $dL/d\varphi$，函数 **ContourFlat** 用来计算对应 φ 的 R_x，R_y，C_x 和 C_y。程序如下：

```
function FlatFacedCam
bate = 60 * pi/180; rb = 3;
h = 0.5; rc = 0.5; n = 23;
phi = linspace(0, beta, n);
ph = [phi, beta + phi];
[Rx, Ry, Cx, Cy] = ContourFlat(phi, rb, h, beta, rc);
ang = linspace(2 * beta, 2 * pi, 40);
plot(Rx, Ry, 'k', rb * cos(ang), rb * sin(ang), 'k')
hold on
plot(Cx(1:5:2 * n), Cy(1:5:2 * n), 'k+')
axis equal
phd = linspace(0, 2 * pi, 50);
[x, phx] = meshgrid(Cx(1:5:2 * n), phd);
y = meshgrid(Cy(1:5:2 * n), phd);
hold on
plot(x + rc. * cos(phx), y + rc. * sin(phx), 'k--')
function[L, dLdphi] = CamProfile(phi, rb, h, beta)
arg = 2 * pi * phi/beta;
L = rb + h * (phi/beta-sin(arg)/2/pi);
dLdphi = (h/beta) * (1-cos(arg));
L = [L fliplr(L)];
dLdphi = [dLdphi, -dLdphi];

function[Rx, Ry, Cx, Cy] = ContourFlat(phi, rb, h, beta, rc)
[L, dLdphi] = CamProfile(phi, rb, h, beta);
theta = atan2(dLdphi, L);
R = L./cos(theta);
ph = [phi, beta + phi];
Ry = R. * sin(theta + ph);
Rx = R. * cos(theta + ph);
gama = atan(dLdphi./(L + rc));
C = (L + rc)./cos(gama);
Cy = C. * sin(gama + ph);
Cx = C. * cos(gama + ph);
```

程序运行后的结果如图 8.28 所示。

8.5.3　偏轴滚子从动件

参考图 8.29 得到以下关系[①]。凸轮轮廓在 (x, y) 的坐标是 R_x 和 R_y，刀具的坐标是 C_x 和 C_y：

$$R_x(\theta, \varphi) = R\cos(\Psi + \varphi + \gamma) \qquad R_y(\theta, \varphi) = R\sin(\Psi + \varphi + \gamma)$$

$$C_x = C\cos(\Psi + \varphi + \delta) \qquad C_y = C\sin(\Psi + \varphi + \delta)$$

① 参见 A.G.Erdman 和 G.N.Sandor，出处同上，第 389～393 页和 H.A.Rothbart，出处同上，第 120～122 页。

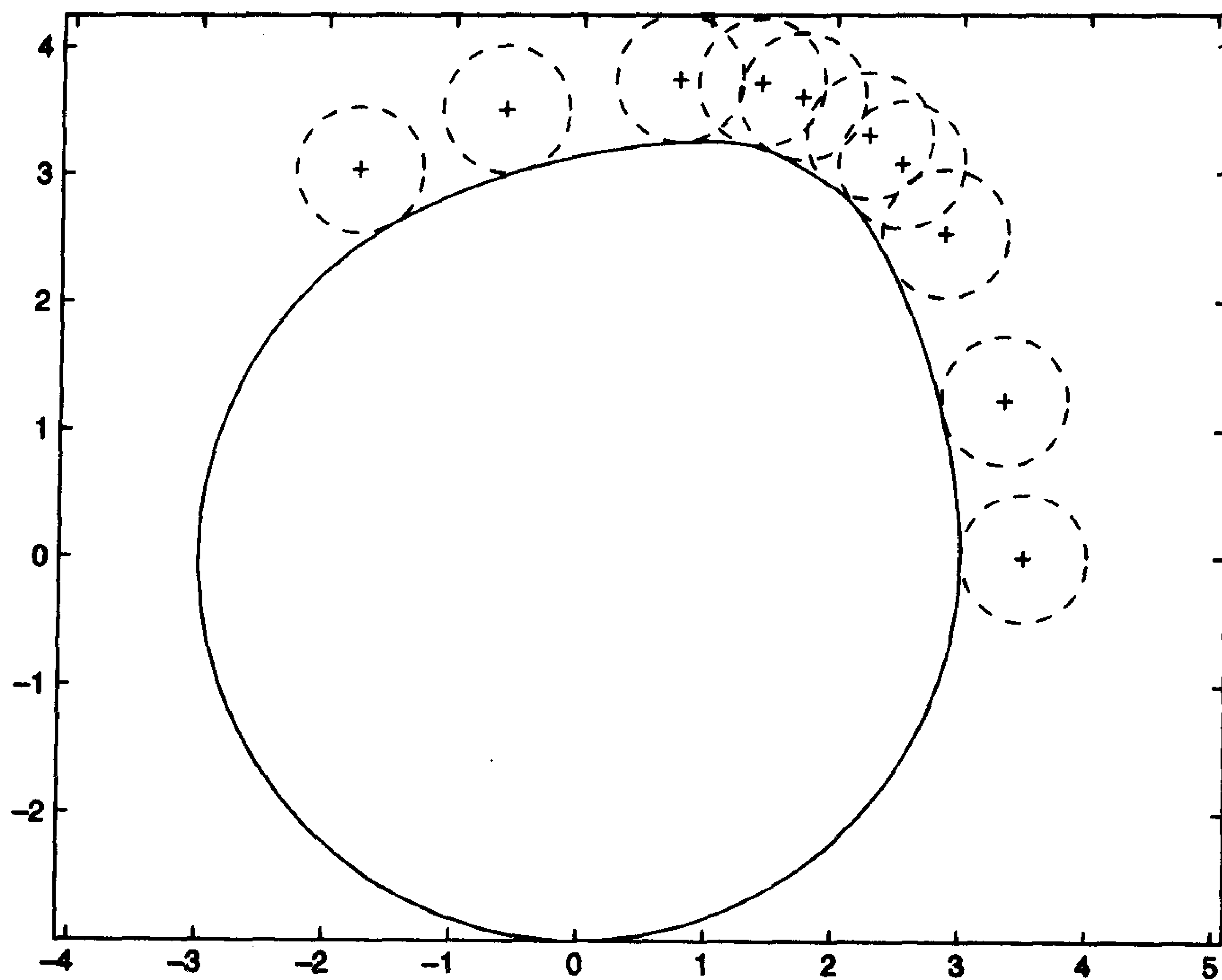


图 8.28 相对于平面从动件摆线运动的凸轮轮廓及其刀具位置

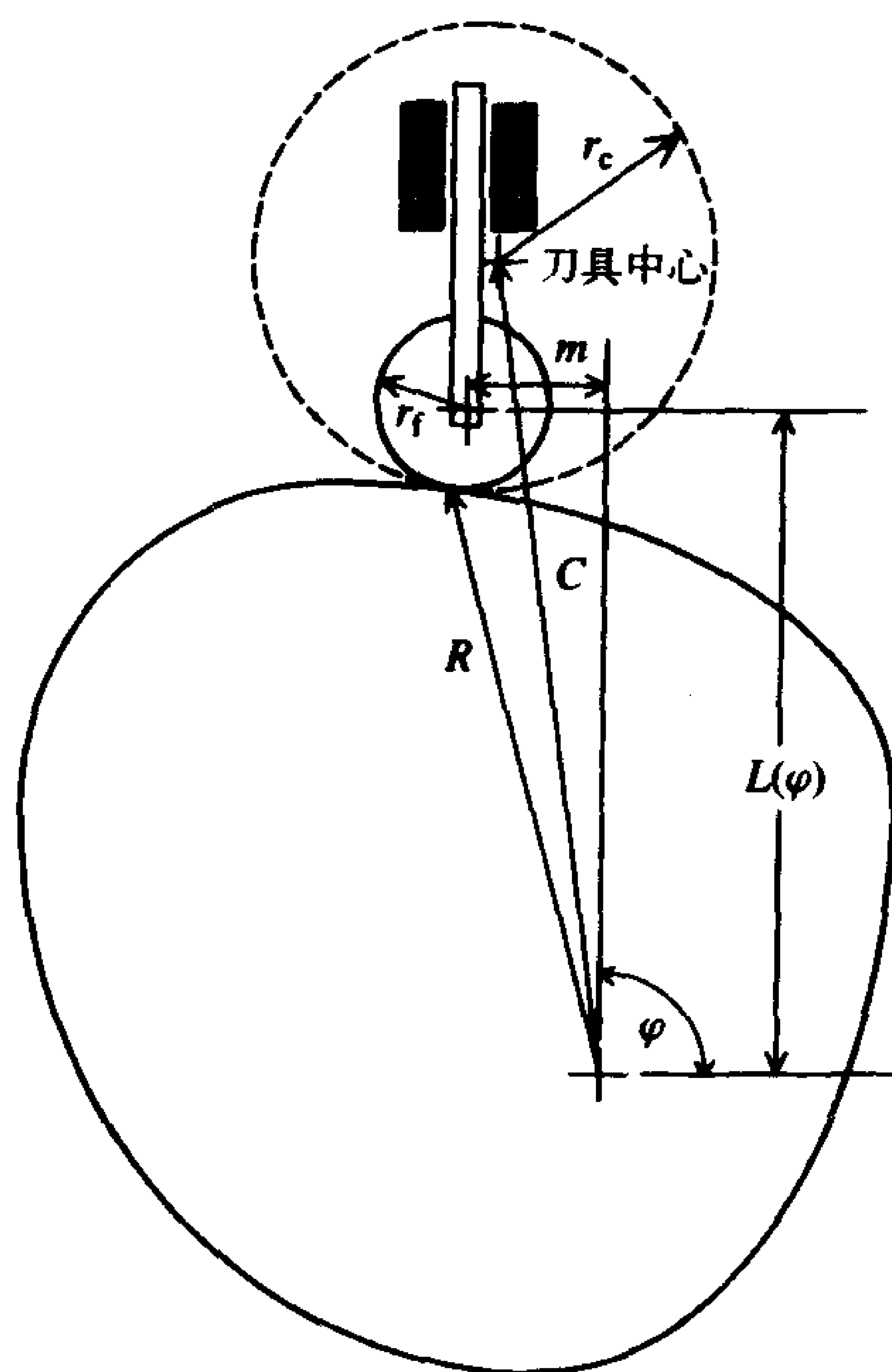


图 8.29 有偏轴滚子从动件的凸轮

其中,

$$R^2 = (F - r_f\cos\alpha)^2 + r_f^2\cos^2\alpha \qquad \Psi = \tan^{-1}(m/L)$$

$$C^2 = c_x^2 + c_y^2 \qquad \alpha = \tan^{-1}\left(\frac{LdL/d\varphi}{F^2 - mdL/d\varphi}\right)$$

$$c_x = F + (r_c - r_f)\cos\alpha \qquad \gamma = \tan^{-1}\left(\frac{F - r_f\cos\alpha}{r_f\sin\alpha}\right)$$

$$c_y = (r_c - r_f)\sin\alpha \qquad \delta = \tan^{-1}(c_y/c_x)$$

$$F^2 = m^2 + L^2$$

凸轮的基圆半径是:

$$L(0) = r_b = \sqrt{R_x^2(\theta,0) + R_y^2(\theta,0)}$$

从 $\varphi = 2\beta + \Delta$ 开始,其中,

$$\Delta = \tan^{-1}\left(\frac{R_y(\theta,0)}{R_x(\theta,0)}\right)$$

通过下例来说明这些结果。

例 8.24　偏轴滚子从动件的凸轮轮廓和刀具的坐标系

当 $\beta = 60°, r_b = 3.0,\ h = 0.5, r_c = 0.5, r_f = 0.375$ 和 $m = 0.375$ 时,计算 R_x, R_y, C_x 和 C_y,并绘出图形。子函数 **ContourRoller** 用来计算对应 φ 的 R_x, R_y, C_x 和 C_y,子函数 **CamProfile** 用来计算 φ 值对应的 $L(\varphi)$和 $dL/d\varphi$(**CamProfile** 与例 8.23 使用相同的子函数)。程序如下:

```
function RollerCam
bate = 60 * pi/180;rb = 3;h = 0.5;rc = 0.5;
rf = 0.375;m = .375;n = 23;
phi = linspace(0.beta,n);
ph = [phi,beta + phi];
[Rx,Ry] = ContourRoller(0,rb,h,beta,rc,m,rf);
rb = sqrt(Rx(1)^2 + Ry(1)^2);
delta = atan2(Ry(1),Rx(1));
[Rx,Ry,Cx,Cy] = ContourRoller(phi,rb,h,beta,rc,m,rf);
ang = linspace(2 * beta + delta,2 * pi + delta,40);
plot(Rx,Ry,'k',Rx(1) * cos(ang),Rx(1) * sin(ang),'k')
hold on
plot(0,0,'k+',Cx(1:5:2 * n),Cy(1:5:2 * n),'k+')
axis equal
phd = linspace(0,2 * pi,50);
[x,phx] = meshgrid(Cx(1:5:2 * n),phd);
y = meshgrid(Cy(1:5:2 * n),phd);
hold on
plot(x + rc. * cos(phx),y + rc. * sin(phx),'k--')

function[Rx,Ry,Cx,Cy] = ContourRoller(phi,rb,h,beta,rc,m,rf)
[L,dLdphi] = CamProfile(phi,rb,h,beta);
F2 = m^2 + L.^2;
F = sqrt(F2);
psi = atan2(m,L);
alpha = atan2(L. * dLdphi,F2-m * dLdphi);
gamma = atan2(rf * sin(alpha),F-rf * cos(alpha));
ph = [phi,beta + phi];
R = sqrt((F-rf * cos(alpha)).^2 + (rf * sin(alpha)).^2);
Ry = R. * sin(psi + gamma + ph);
```

```
Rx = R. * cos(psi + gamma + ph);
cx = F + (rc-rf) * cos(alpha);
cy = (rc-rf) * sin(alpha);
delta = atan2(cy,cx);
C = sqrt(cx.^2 + cy.^2);
Cy = C. * sin(psi + delta + ph);
Cx = C. * cos(psi + delta + ph);
```

程序执行结果如图 8.30 所示。

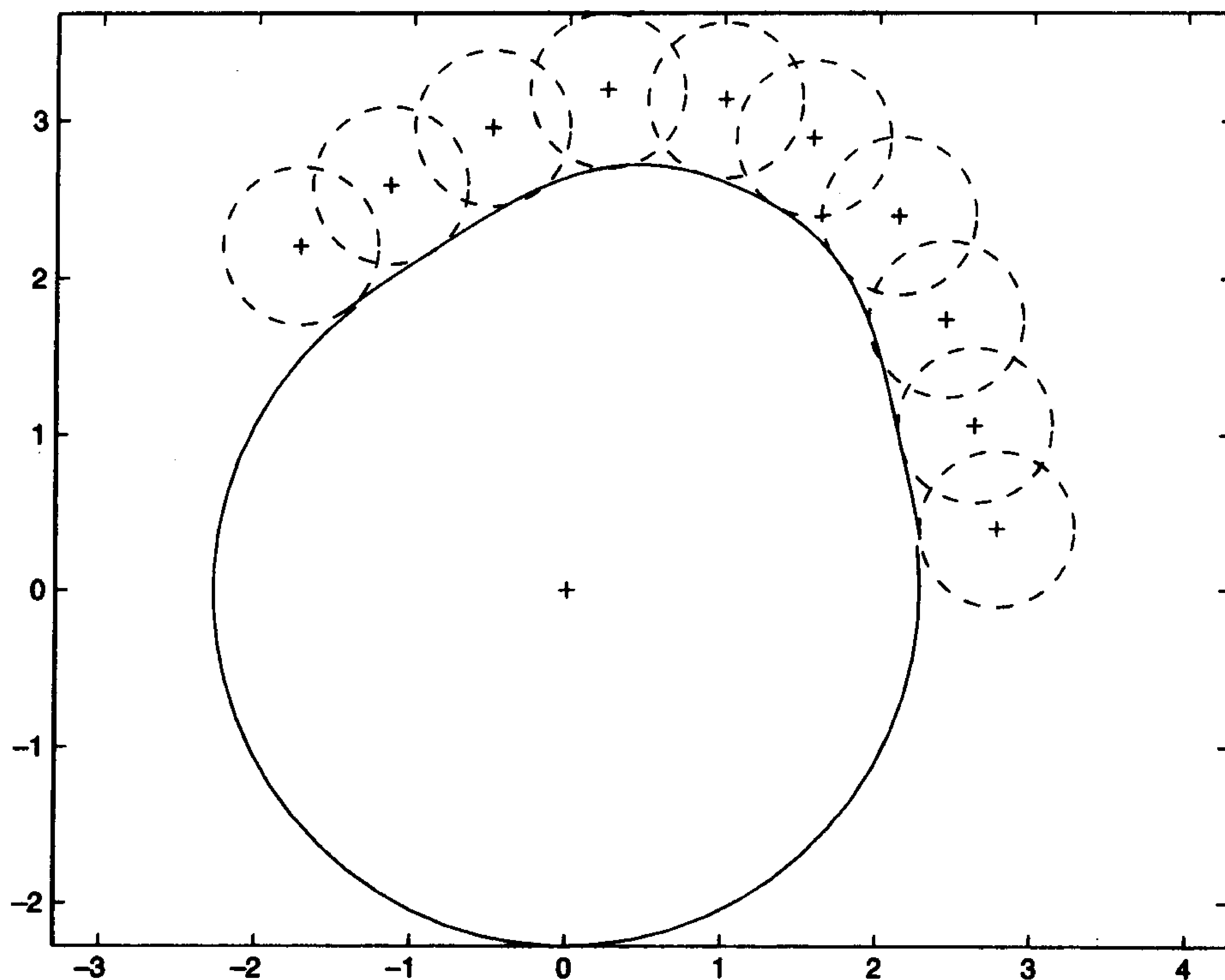


图 8.30 相对于偏轴滚子从动件摆线运动的凸轮轮廓及其刀具坐标

8.5.4 凸轮的曲率半径

凸轮轮廓的曲率半径给出如下：

$$\rho = \frac{[(L(\varphi))^2 + (\mathrm{d}L(\varphi)/\mathrm{d}\varphi)^2]^{3/2}}{(L(\varphi))^2 + 2(\mathrm{d}L(\varphi)/\mathrm{d}\varphi)^2 - L(\varphi)\mathrm{d}^2L(\varphi)/\mathrm{d}\varphi^2}$$

凸轮轮廓应该是这样的：从动件的曲率半径总是大于凸轮轮廓的最小曲率半径，有意义的是最小曲率半径。使用 8.5.1 节中对无量纲位移、速度、加速度和瞬时加速度的定义，曲率半径可表示如下：

$$\rho/h = \frac{[(r_b/h + S)^2 + V^2]^{3/2}}{(r_b/h + S)^2 + 2V^2 - (r_b/h + S)A}$$

通过下例来说明这些结果。

例 8.25　最小凸轮曲率半径

对于任意的 r_b/h 和 β 值,计算最小曲率半径(无量纲)。子函数 **CamCurvature** 利用曲率半径与 β 的对称关系来计算 ρ,其中 $0 \leqslant \varphi \leqslant \beta$。程序如下:

```
function CamRadiusCurvature
rbh = input('Enter ratio rb/h:');
beta = input('Enter angle beta(degrees):') * pi/180;
options = optimset('display','off');
phimin = fminbnd(@CamCurvature,0,beta,options,beta,rbh);
rmin = CamCurvature(phimin,beta,rbh);
disp(['When beta = 'num2str(beta * 180/pi)'degrees and rb/h = '...
      num2str(rbh)'the minimum radius of curvature for a'])
disp(['cycloidal cam profile is = 'num2str(rmin)'h,which occurs at'...
num2str(phimin * 180/pi)'degrees.'])
function RadiusCurve = CamCurvature(phi,beta,rbh)
arg = 2 * pi * phi/beta;
S = phi/beta-sin(arg)/2/pi;
V = (1-cos(arg))/beta;
A = 2 * pi/beta^2 * sin(arg);
RadiusCurve = ((rbh + S)^2 + V^2)^1.5/((rbh + S)^2 + 2 * V^2 - (rbh + S) * A);
```

程序执行后,在 MATLAB 命令窗口中显示如下信息:

```
Enter ratio rb/h:4
Enter angle beta(degrees):80
When beta = 80 degrees and rb/h = 4 the minimum radius of curvature for a
cycloidal cam profile is = 2.9777h,which occurs at 58.8421 degrees.
```

其中,用户输入值是 4 和 80。

8.6　滑动轴承

考虑如图 8.31 所示的短颈轴承。如果 $c_d = (d_b - d_j)$,那么径向间隔 $c_r = c_d/2$ 是偏心距 e 的最大值,偏心率定义为 $\varepsilon = e/c_r$。然而,实际上这个值通常由下面的经验公式获得:

$$\varepsilon \rightarrow \varepsilon_x = 0.213\,94 + 0.385\,17 \lg O_N - 0.0008(O_N - 60)$$

其中,O_N 是一个载荷系数,或称为 Ocvirk 数,由下式给出:

$$O_N = \frac{P}{nLd\eta}\left(\frac{d}{L}\right)^2\left(\frac{c_d}{d}\right)^2 = 4\pi K_\varepsilon$$

其中,η 是油的绝对黏度,n 是轴颈旋转速度,以每秒运行的圈数来记。设计目标是保持 $O_N < 30$。

最小的油膜厚度给出如下:

$$h_{\min} = c_r(1 - \varepsilon_x)$$

为了最大程度减小表面接触的机会,$h_{\min}$应该是比轴承和轴颈表面光洁度大 2 到 3 倍的系数。

轴承能承受的载荷是 :

$$P = 4\pi K_\varepsilon \eta n d L^3 / c_d^2$$

该转动轴转矩为:

$$T_r = \frac{\pi^2 d^3 L n \eta}{c_d \sqrt{(1 - \varepsilon_x^2)}} + 0.5 P \varepsilon_x c_d \sin\phi$$

其中,在等号右边的第一项是稳定转矩,而:

$$\phi = \tan^{-1}\left(\frac{\pi\sqrt{1-\varepsilon_x^2}}{4\varepsilon_x}\right)$$

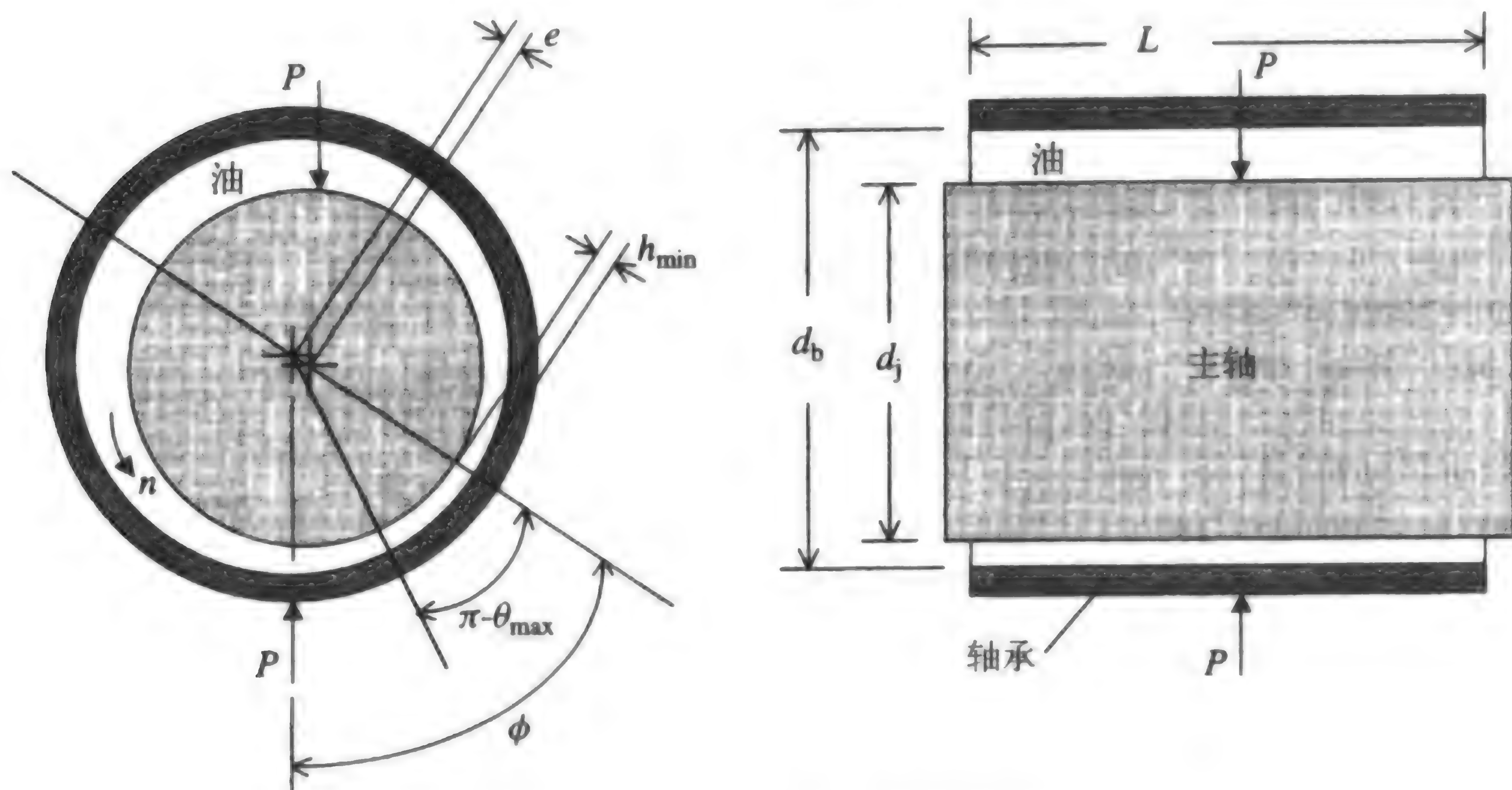


图 8.31 关于滑动轴承的术语

轴颈和轴承之间的摩擦系数是:

$$\mu = \frac{2T_r}{Pd}$$

油的华氏温度的增加值给出如下:

$$\Delta T = \frac{2\pi n T_r}{6600\rho c_p Q_H}$$

其中,

$$Q_H = \pi dnL\varepsilon_x c_d/2$$

Q_H 是油的流量,单位是 in^3/s;润滑剂的热容量为 C_p;ρ 是其密度。对于润滑油,有近似值 c_p = 0.48 Btu/(lb-℉)和 ρ = 0.031 lb/in^3。假设温度升高是从入口油的温度 T_{in} 开始的。因为 η(见下面)和 T_r 是温度的函数,所以这个方程需要迭代求解。使用 c_p 和 ρ,得到:

$$\Delta T = \frac{0.0640 n T_r}{Q_H}$$

油的绝对黏度通过以下关系可以在 ± 10% 之内估计,单位是雷恩(lb-s/in^2):

$$\eta = 10^{C-7}$$

其中,

$$C = 10^{A_j - B_j \lg T_o}$$

其中,$T_o = 255.2 + 5/9T$ K,T 是油的华氏温度,A_j 和 B_j 在表 8.6 中给出,是油的 SAE 值的函数(参见习题 6.11)。

表 8.6 用于确定 η 的常数

SAE 数	j	A_j	B_j
10	1	9.1209	3.5605
20	2	9.1067	3.5385

(续表)

SAE 数	j	A_j	B_j
30	3	8.9939	3.4777
40	4	8.9133	3.4292
50	5	8.5194	3.2621
60	6	8.3666	3.1884

通过以下例子来说明这些结论。

例 8.26 滑动轴承的载荷容量

考虑以下滑动轴承的指标:

油的入口温度 = 120 ℉

SAE 30 油

轴转速 = 3000 r/m

轴直径 = 2.0 in

主轴长 = 1.1 in

环向间隙 = 0.0013 in

当 Ocvirk 数低于 25 时,确定轴承能承受的最大载荷 P,以及相应的温升、最小的油膜薄片厚度和绝对黏度值。

首先,创建函数 **JournalLoad**,对给定载荷计算油的温升、绝对黏度和 Ocvirk 数。这个函数由以下三个子函数支持:

AbsViscosity—计算绝对黏度 η

Ocvirk—计算 Ocvirk 的 O_N

TempRise—计算温升 ΔT

然后,使用 fzero 命令来确定载荷 P 的最大值,当 P 为最大值时 Ocvirk 值为 25,函数文件如下:

```
function[Omax,absvis,ex,temp] = JournalLoad(P)
global Tin SAE n d cd L
temp = Tin;Told = 0;
while abs(temp-Told) > 0.1
  Told = temp;
  absvis = AbsViscosity(temp);
  ex = Ocvirk(absvis,P);
  deltaT = TempRise(temp,ex,P);
  temp = Tin + deltaT;
end
absvis = AbsViscosity(temp);
[ex,ocv] = Ocvirk(absvis,P);
Omax = 25-ocv;

function absvis = AbsViscosity(tempF)
global SAE
AandB = [9.1209 3.5605;9.1067 3.5385; 8.9939 3.4777;8.9133 3.4292;...
        8.5194 3.2621;8.3666 3.1884];
SAE10 = SAE/10;
```

```
absvis = 10^(10^(AandB(SAE10,1)-AandB(SAE10,2) * log10(255.2...
              + 5/9 * tempF))-7);

function[ex,ocv] = Ocvirk(vis,P)
global n d cd L
ocv = P/n/L/d/vis * (cd/L)^2;
ex = 0.21394 + 0.38517 * log10(cov) - 0.0008 * (ocv - 60));

function deltaT = TempRise(temp,ex,P)
global Tin SAE n d cd L
QH = pi * d * (n) * L * ex * cd/2;
phi = atan(pi * sqrt(1 - ex^2)/4/ex);
vis = AbsViscosity(temp);
Tr = pi^2 * d^3 * L * (n) * vis/cd/sqrt(1 - ex^2) + 0.5 * P * ex * cd * sin(phi);
deltaT = (0.0640) * Tr * (n)/QH;
```

程序如下:

```
global Tin SAE n d cd L
Tin = 120;SAE = 30;n = 3000/60;
d = 2;cd = .0026;L = 1.1;
options = optimset('display','off');
P = fzero(@JournalLoad,1000,options);
[Omax,absvis,ex,temp] = JournalLoad(P);
disp(['Maximum load = 'num2str(P)'lb.'])
disp(['Temperature rise = 'num2str(temp-Tin)'degrees F'])
disp(['Absolute viscosity = 'num2str(absvis)'reyn'])
disp(['Minimum oil film thickness = 'num2str(0.5 * cd * (1 - ex))'inches'])
```

执行程序后,在MATLAB命令窗口中显示以下内容:

```
Maximum load = 1078.3427 lb.
Temperature rise = 58.6577 degrees F
Absolute viscosity = 2.1907e-006 reyn
Minimum oil film thickness = 0.0002855 inches
```

8.7 PDE工具箱和薄板凹槽处的应力集中系数

MATLAB的偏微分方程(PDE, Partial Differential Equation)工具箱[①]及其图形用户界面(GUI, graphical user interface)一起用来分析几类二维场方程。一类是平面应力问题。平面应力问题是一种近似方法,用于确定任意形状和厚度 h 的平面薄板的应力、拉力和位移。假设应力 σ_z 在 $z=\pm h/2$ 处垂直于平板的表面,并且在平板表面的剪切应力 σ_{xz} 和 σ_{yz} 是零,则其他应力独立于 z。

使用工具箱和GUI在矩形钢板上确定集中应力系数。在平板两边分别开出两个对称于中点的半圆形凹槽,平板的形状和说明如图8.32所示。平板在左端固定,在平板的右端施加沿 x 轴方向上的力,每单位长度上力的大小为 $F_x=20$ N/cm。如果平板的厚度是0.2 cm,那么在 x

① PDE工具箱在流动方面的可视化应用参见本书11.3.3节的例11.8,在传热方面的应用参见本书12.3.3节的例12.7。

轴正方向上的边缘应力是 100 N/cm²。钢板的弹性模量在现在的单位条件下是 200×10^5 N/cm²。

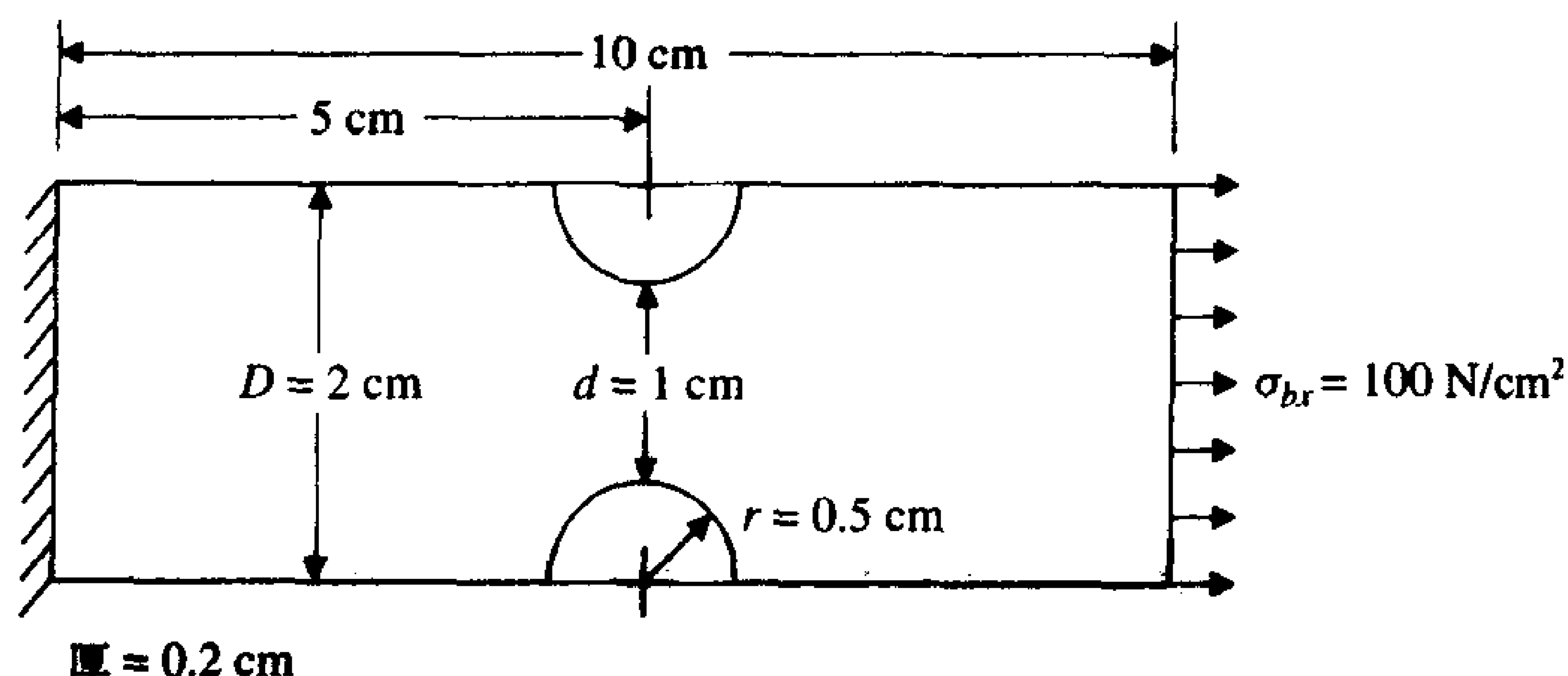


图 8.32 拉伸作用下的开槽平板

PDE 工具箱要求给出每一个边的边界条件，在平板应力问题中通常需要考虑两种类型的边界条件：(1)位移是零或是给定值，(2)表面应力是零或是给定值。位移的给定值在 GUI 中描述为第立克特(Dirichlet)边界条件，而应力的给定值描述为纽曼(Neumann)边界条件。

在(x,y)坐标系中的任意方向上，研究作用在边缘上的两个应力。使用 GUI 注释，如果作用在 x 方向上的应力分量是g_1，作用在 y 方向上应力分量是g_2，则：

$$g_1 = n_x\sigma_{bx} + n_y\sigma_{bxy}$$

$$g_2 = n_y\sigma_{by} + n_x\sigma_{bxy}$$

其中，σ_{bx}是x 方向上的边界应力，σ_{by}是y 方向上的边界应力，σ_{bxy}是边界剪切应力。另外，n_x和n_y分别是x 轴和y 轴的方向余弦。这样，在与 x 轴平行的边界上，$n_x = 0$，$n_y = \pm 1$($+1$ 表示与 y 轴同向)，规定表面应力为：

$$g_1 = \pm \sigma_{bxy}$$

$$g_2 = \pm \sigma_{by}$$

如果边缘部分的剪切应力是零，即 $\sigma_{bxy} = 0$，则 $g_1 = 0$；相反，如果法向应力 $\sigma_{by} = 0$，则 $g_2 = 0$。如果边缘部分无应力，则 $g_1 = g_2 = 0$。

当处理弯曲边缘(圆弧或椭圆弧)时，可以规定边界条件如下：

$$g_1 = N * nx$$

$$g_2 = N * ny$$

其中，nx 和 ny 由 GUI 分别表示为相对于 x 轴和y 轴的方向余弦，并且 N 是一个数字值，用于表示弯曲边界的法向应力值。

使用 PDE GUI 来确定应力、拉力和位移的过程有如下几步：

1. 设置画图面积；
2. 选择合适的 PDE——在这种情况下应选平面应力；
3. 画出(或创建)二维形状图；
4. 规定边界条件；
5. 选择物理量；
6. 生成格状图；
7. 求解；
8. 显示结果；

9. 如果需要进一步分析,则将 GUI 生成的数组输出到 MATLAB 的命令窗口中去。

下面给出这些步骤的实现细节。在这一过程中任何一个步骤都能返回到前面已完成的步骤并进行改变,那么前面已完成的步骤也必须能再次执行。为了访问 GUI,在 MATLAB 命令窗口中输入:

```
pdetool
```

这是一个 MATLAB 函数。它打开 PDE 工具箱的 GUI,如图 8.33 所示。

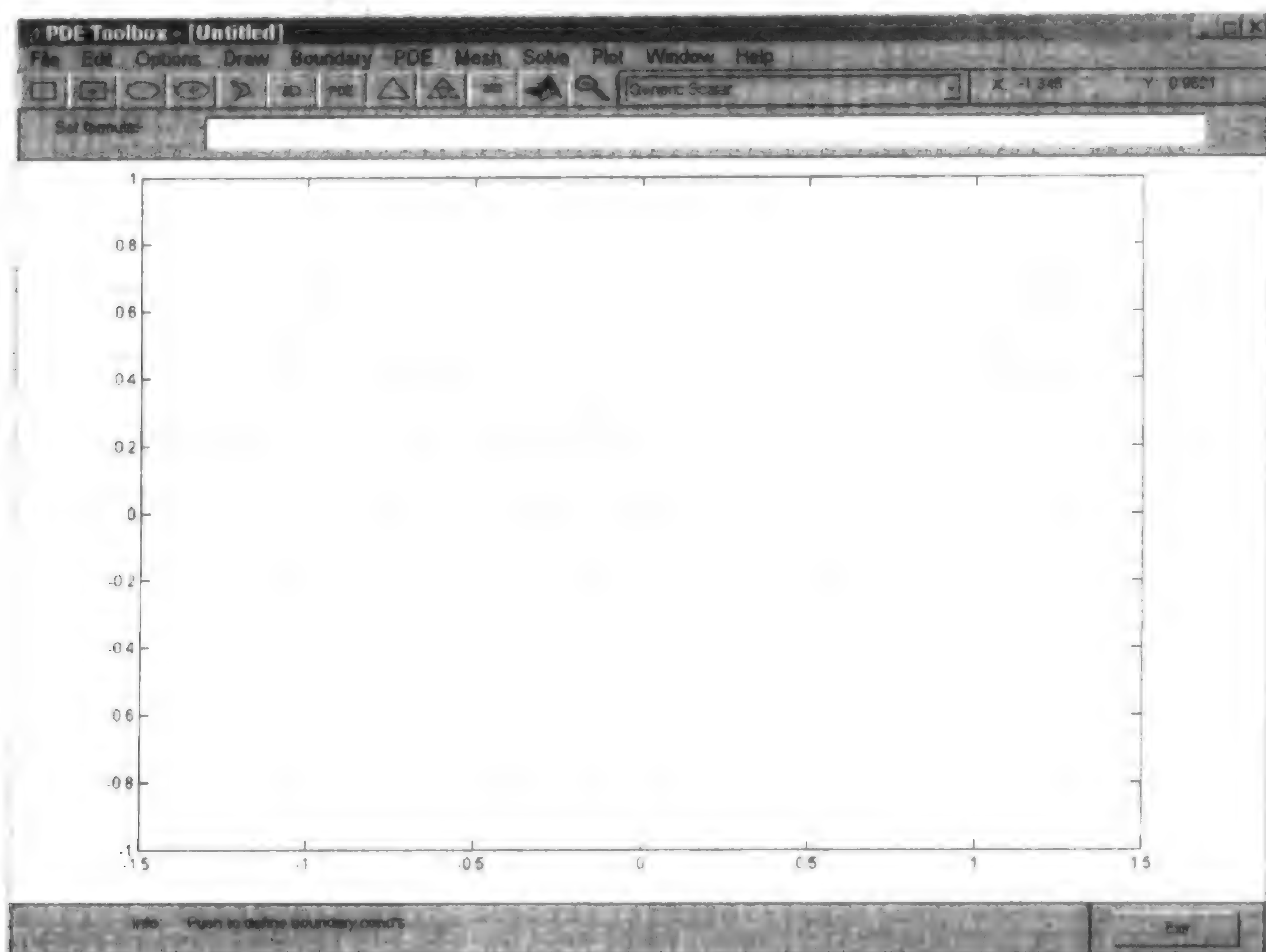


图 8.33 PDE 工具箱的 GUI

设置画图面积

基于图 8.32 中对平板的说明,创建一块 12×4 单位的绘图面积,其中 1 单位 = 1 cm。然后显示网格线,且通过点击 *Snap* 出现"整格移动"选项,此后多次点击 *Options* 菜单。第一次点击 *Grid*,出现检查记号(√)。然后在 *Snap* 上做同样的操作。接下来点击 *Axis Limits*。在菜单窗口中,输入 X 轴范围:

[0 12]

输入 Y 轴范围:

[0 4]

然后点击 *Apply* 以及 *Close*。先点击 *Apply* 确定 *Auto* boxes 是空的。最后,点击 *Grid Spacing* ,输入 X 轴的线性分割为:

0:0.5:12

Y 轴的线性分割为:

0:0.5:4

将该面积标记为空,确保 *Auto* boxes 也是空的。点击 *Apply*,然后点击 *Done*,这些选择的结果如图 8.34 所示。

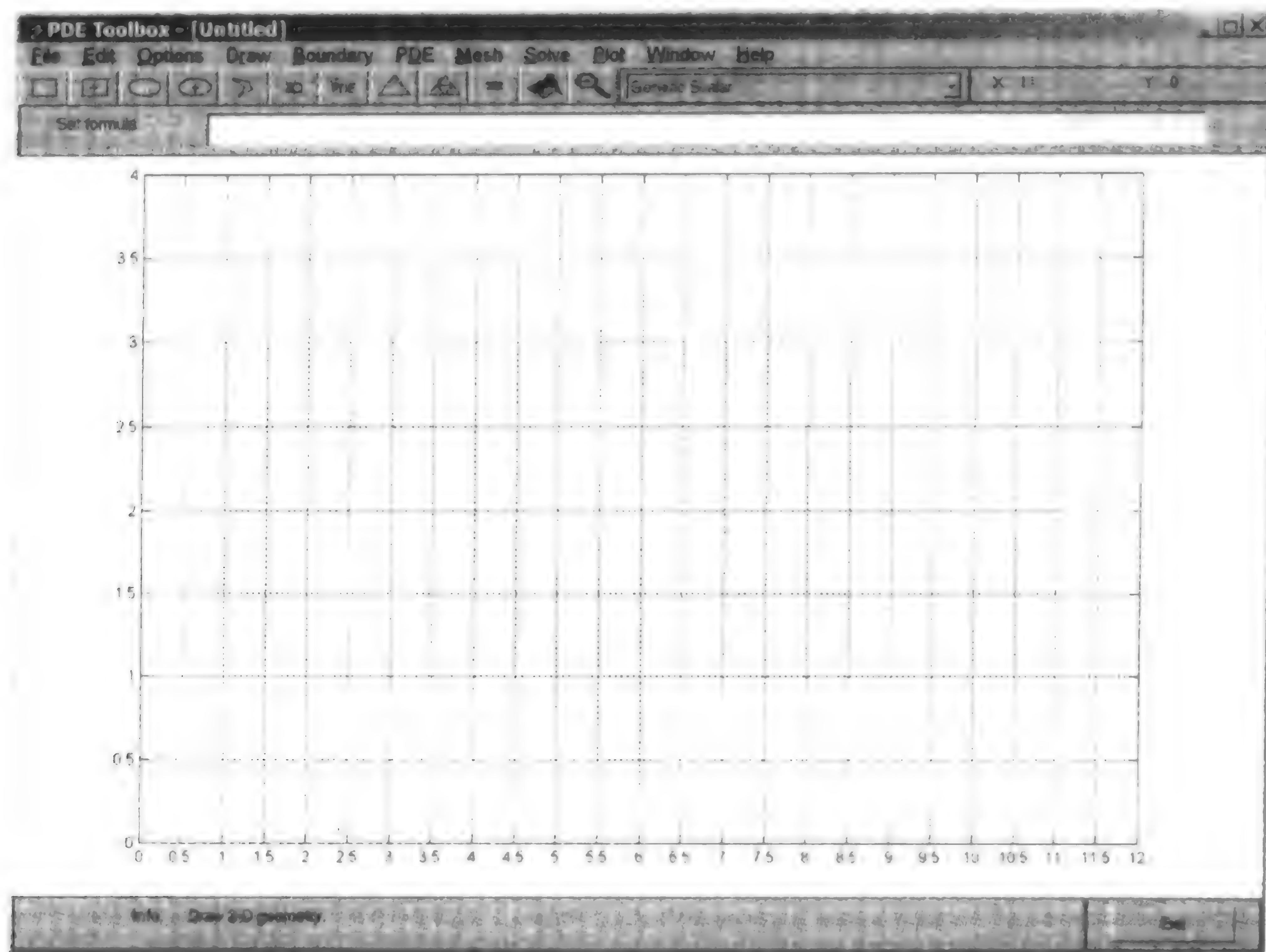


图 8.34　设置画面面积的结果

选择 PDE

为了选择 PDE,进入工具条的右端,点击指向描述符 *Generic Scalar* 旁的箭头。当 pdetool 打开后,该描述符就能显示出来,然后选择 *Structural Mech* 和 *Plane Stress*。

画出(创建)二维形状图

绘图程序采用积木方法创建平面图形,通过对矩形(正方形)、椭圆(圆)和多边形的任意组合进行填加和删除,从而完成绘图过程。每一个图元都能放置在已创建区域的任何地方,并且能加入说明,但不能超过所画区域的边缘。十字光标的位置坐标在 GUI 窗口的右上角给出。点击绘图区域左上角的某种图标即可选中该图元,并将其绘制出来。根据选择方式不同,用鼠标键可按图元的顶角放置,也可按中心放置。然后按下鼠标键,光标移动到下一个位置并松开。*Snap* 选项可简化这种放置过程。

每个图元放置好以后,一个文字数字指示器显示在刚创建的图元中心,同时显示在输入数据窗口,输入数据窗口在绘图区的上端和 *Set formula* 注释区的右侧。约定是:*R* 代表矩形,*SQ* 代表正方形,*E* 代表椭圆,*C* 代表圆,*P* 代表多边形。一个图元放置后其文字数字指示器的符号设定为正号(+)。要从一个图中移走(或删除)一个图元,应进入到数据区域改变相应图元的符号为负号(-)。变化符号以后,按回车键。

通过创建图 8.32 所示的平面图来说明这一步。首先,点击工具条打开一个矩形,然后将光标放置在坐标(1,1)处并点击,按住鼠标键拖动至坐标(11,3)处,放开鼠标键,就得到一个 10×2 的矩形。如果不想要此矩形,那么到 *Edit* 菜单选择 *Clear* 或在键盘上按下 *Del* 键加以清除。当放置了多个图元时,要首先选中被删除的图元,否则所有图元都将被删除。被选中的图元有一个黑色的边框。

接下来选择中心的放置方式,将光标放置在(6,1)处画一个椭圆(圆),按下鼠标键,拖动光标直到出现一个直径为 1 的圆。因为坐标轴是不等的,所以在做图区域中出现一个椭圆,重复这一过程可画出中心为(6,3)的圆。如果不需要网格线,也可以将其关闭。最终结果如图 8.35 所示。

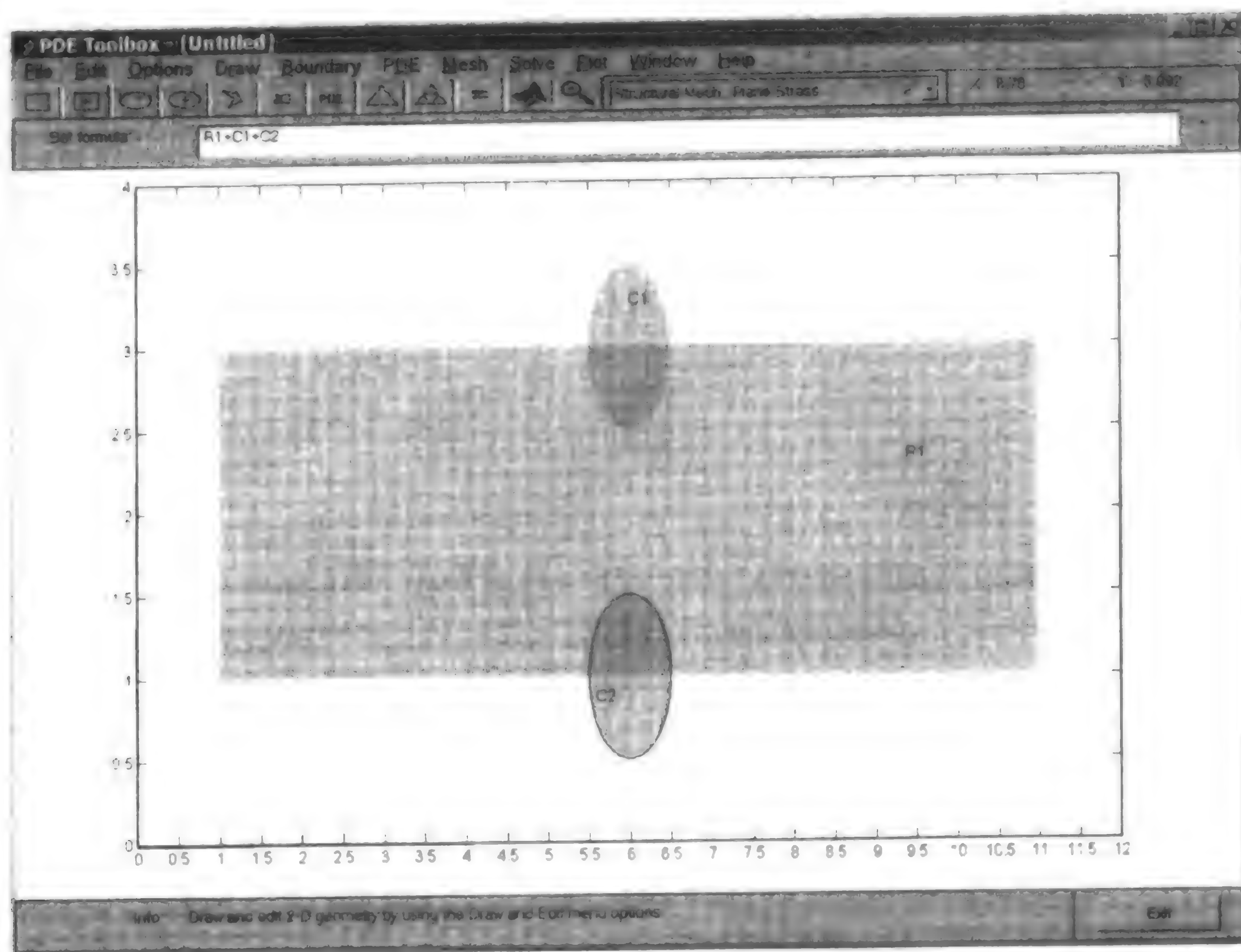


图 8.35　画出一个矩形和两个圆的结果

在 *Set formula* 区域中将

$R1 + C1 + C2$

变为：

$R1 - C1 - C2$

并按回车键。

指定边界条件

在规定每一个边缘部分的边界条件以前，必须让 GUI 执行一系列操作。首先，进入 *Boundary* 菜单并选择 *Boundary Mode* 选项。操作结果如图 8.36 所示。将区域中的图形设置为边界选择模式。

接下来的工作将会围绕每一个边界部分展开，边界部分开始于一个箭头的顶端，结束于另一个相邻箭头的顶端。按任意顺序确定每一部分的边界条件，建议采用顺时针方向或逆时针方向，最大限度保证边界不丢失。在段 # 1 开始，沿图形边界顺时针旋转。

当点击段 # 1 时，它由红变黑。然后双击这一部分，使仅出现该部分的 *Boundary Conditions* 窗口。因为边缘是固定的，所以具有零位移。就像前面所提到的，位移是狄利克雷边界条件。这样，在 *Condition type* 中选择 *Dirichlet*。既然位移是零，可以使用默认的边界条件：$h_{11} = h_{22} = 1$，$h_{12} = h_{21} = r_1 = r_2 = 0$。位移标号出现在菜单窗口的上方，$u$ 代表两个向量分量：一个是 x 方向上的位移，另一个是 y 方向上的位移。这样，r_1 与 x 方向上的位移相关，r_2 与 y 方向上的位移相关。点击 *OK* 按钮关闭边界条件窗口，线段的颜色仍恢复为红色。

因为剩余边缘部分可能无应力作用或有一个已知的应力作用（# 3 部分），所以都是纽曼边界条件。这样对于段 # 2 和所有无应力作用的边缘部分使用以下过程。再次双击该线段，显示 *Condition Type* 选择 *Neumann*，其他值都是默认值，即 0。点击 *OK* 按钮，关闭边界条件窗口。此时线段为蓝色，除线段 # 3 以外其余的线段都可以这样做。对于线段 # 3，再选择 *Neumann*。$g1$ 的输入值是 100，代表 100 N/cm^2。其余的默认值保持不变。

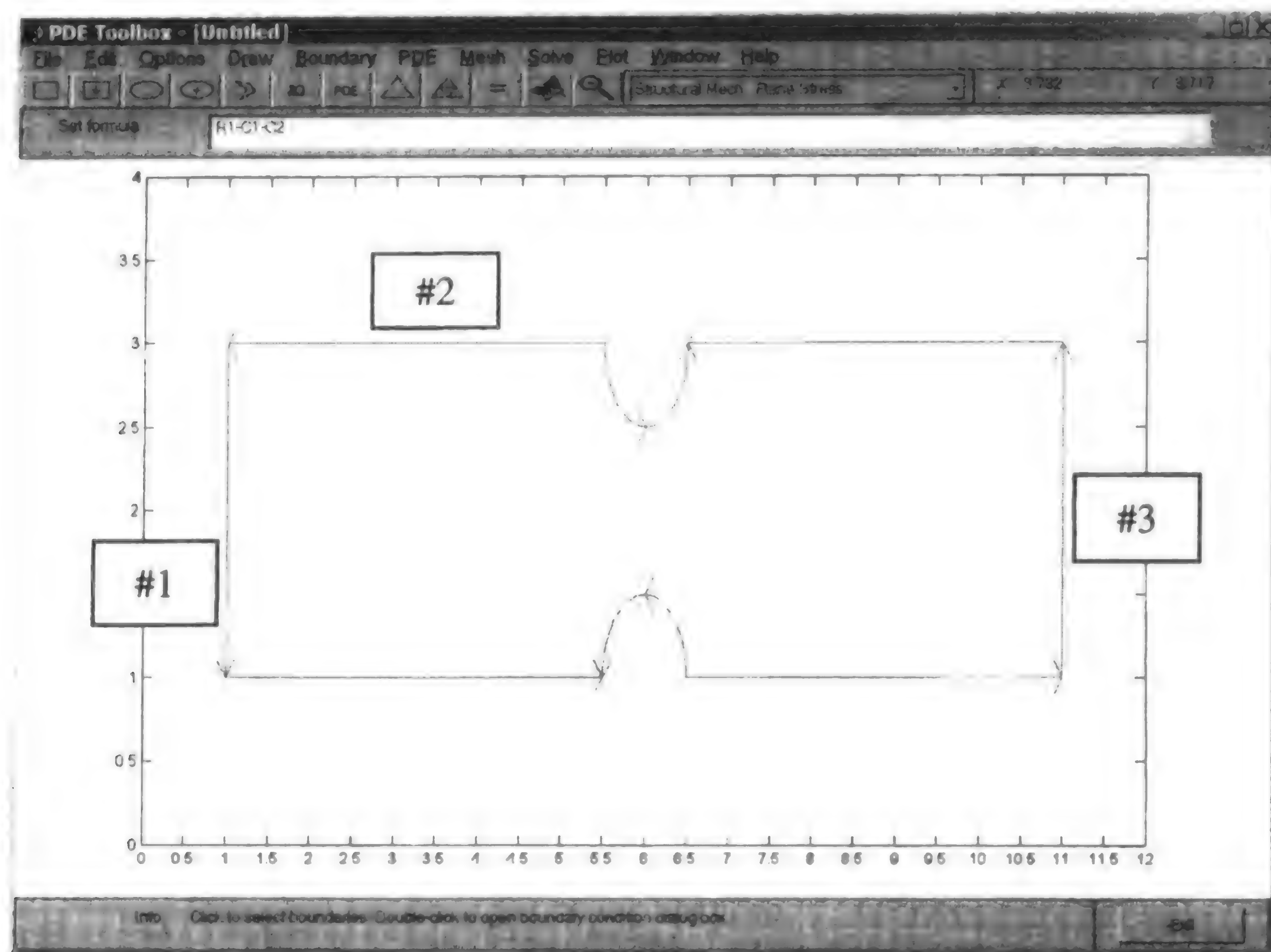


图 8.36　设置成 *Boundary Mode* 操作的结果(注意:图上的数字并不属于显示内容)

指定物理常量

为了指定平板的物理常量,进入 *PDE* 并选择 *PDE Specification* 选项。*E* 值的输入为 200e5,默认值 *nu* = 0.3(泊松比)和 *rho* = 1(质量密度)。因为是静态问题,所以不使用密度,然后点击 *OK* 按钮。

生成格状图

为了让系统生成格状图,从 *Mesh* 菜单中选择 *Initialize Mesh* 选项,结果如图 8.37 所示。如果网格显示过份粗糙,则回到 *Mesh* 菜单并选择 *Refine Mesh* 选项,结果如图 8.38 所示。每选择 *Refine Mesh* 选项一次,格状三角形就变小一次。

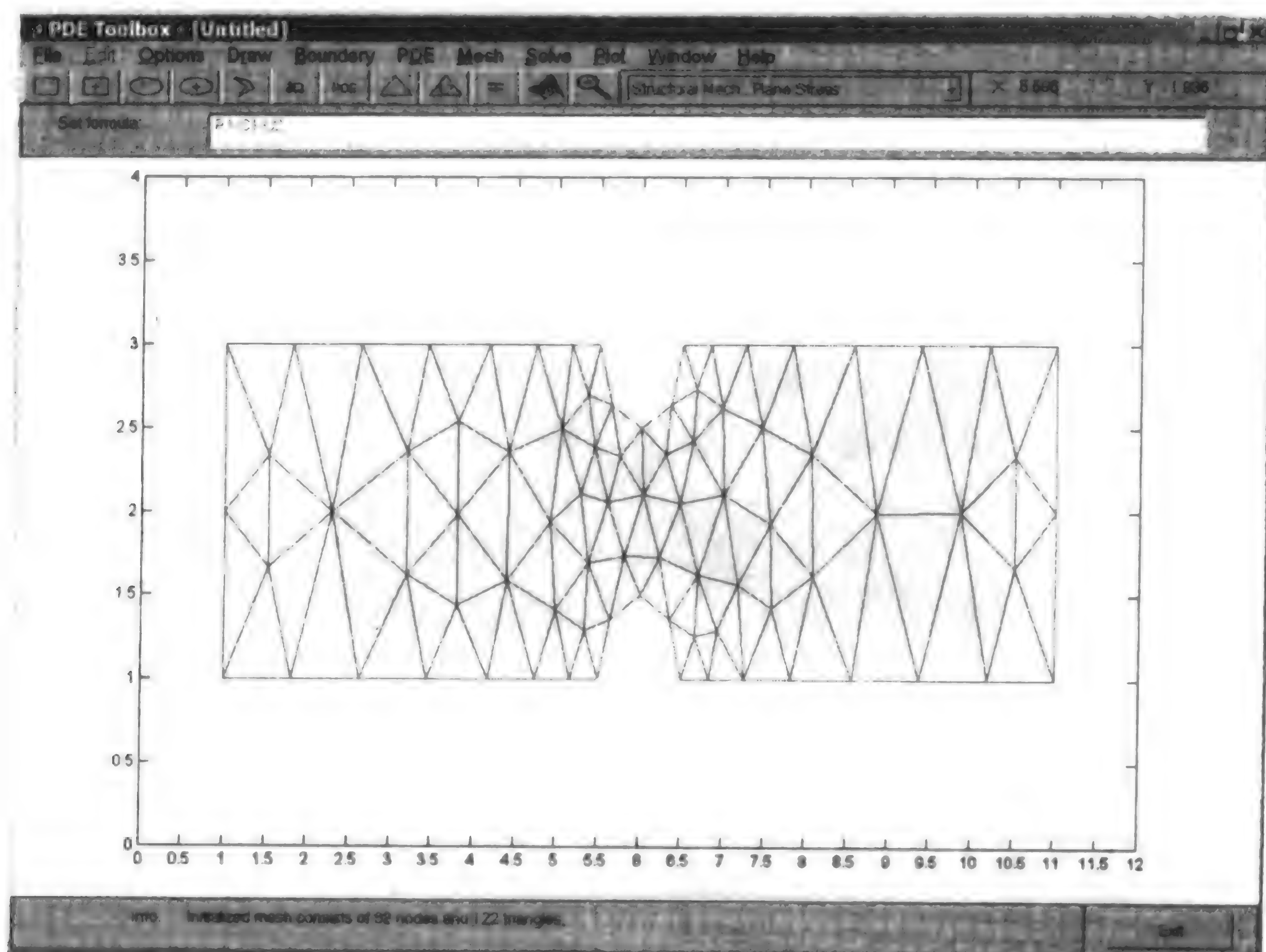


图 8.37　从 *Initialize Mesh* 选项得到的结果

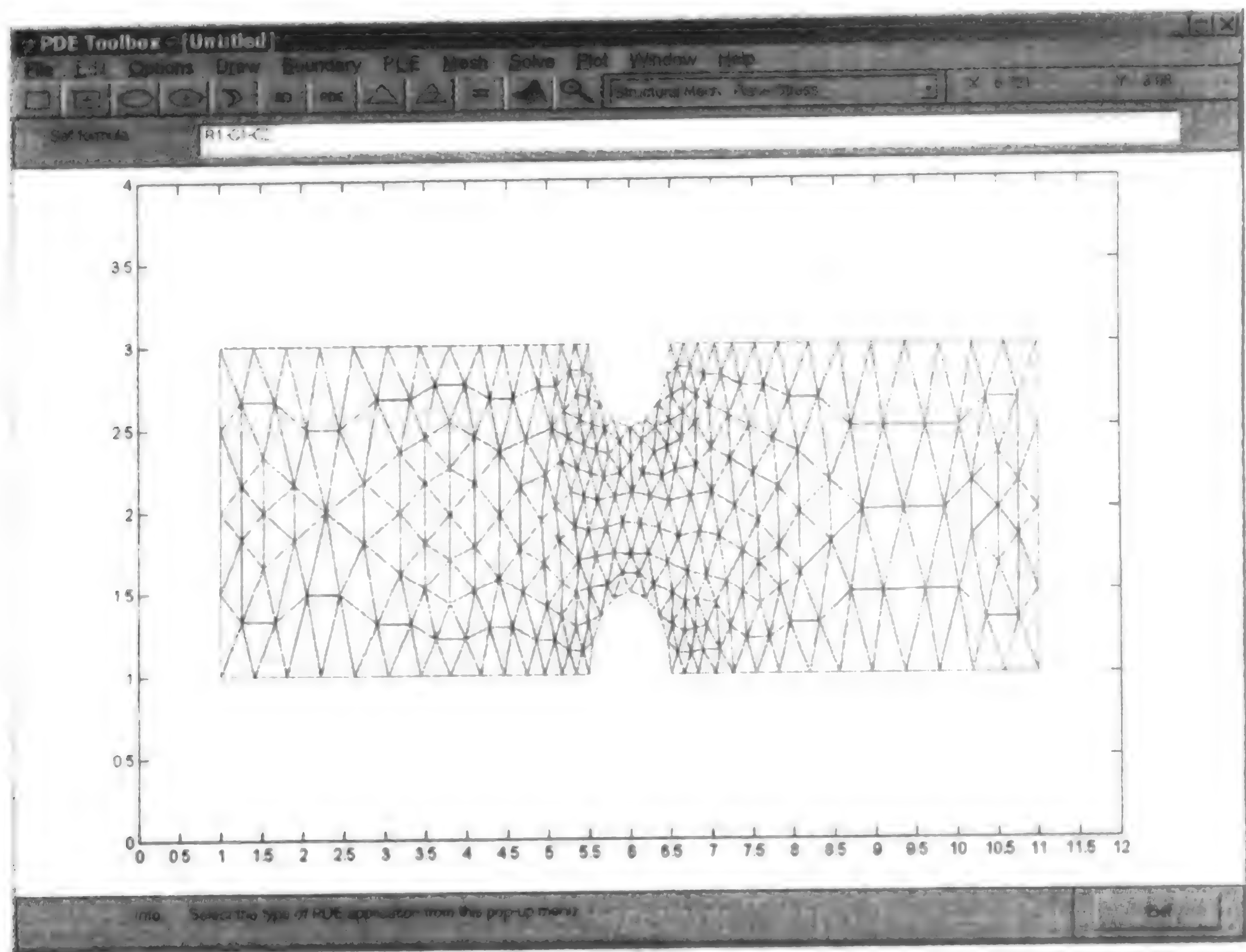


图 8.38 使用一次 *Rifine Mesh* 选项后的结果

求解

求解的方法是选择 *Solve* 菜单中的 *Solve PDE* 选项或者点击“ = ”图标。

显示结果

若以一种形式显示多个结果,则最好对其进行说明。进入 *Plot* 菜单并选择 *Parameters* 选项。σ_{xx} 的分布见图 8.39,在 *Plot Type* 菜单下选择 *Color* 选项和 *Contour* 选项,在 *Property* 的第一个可选框中选择 *x stress* 选项。

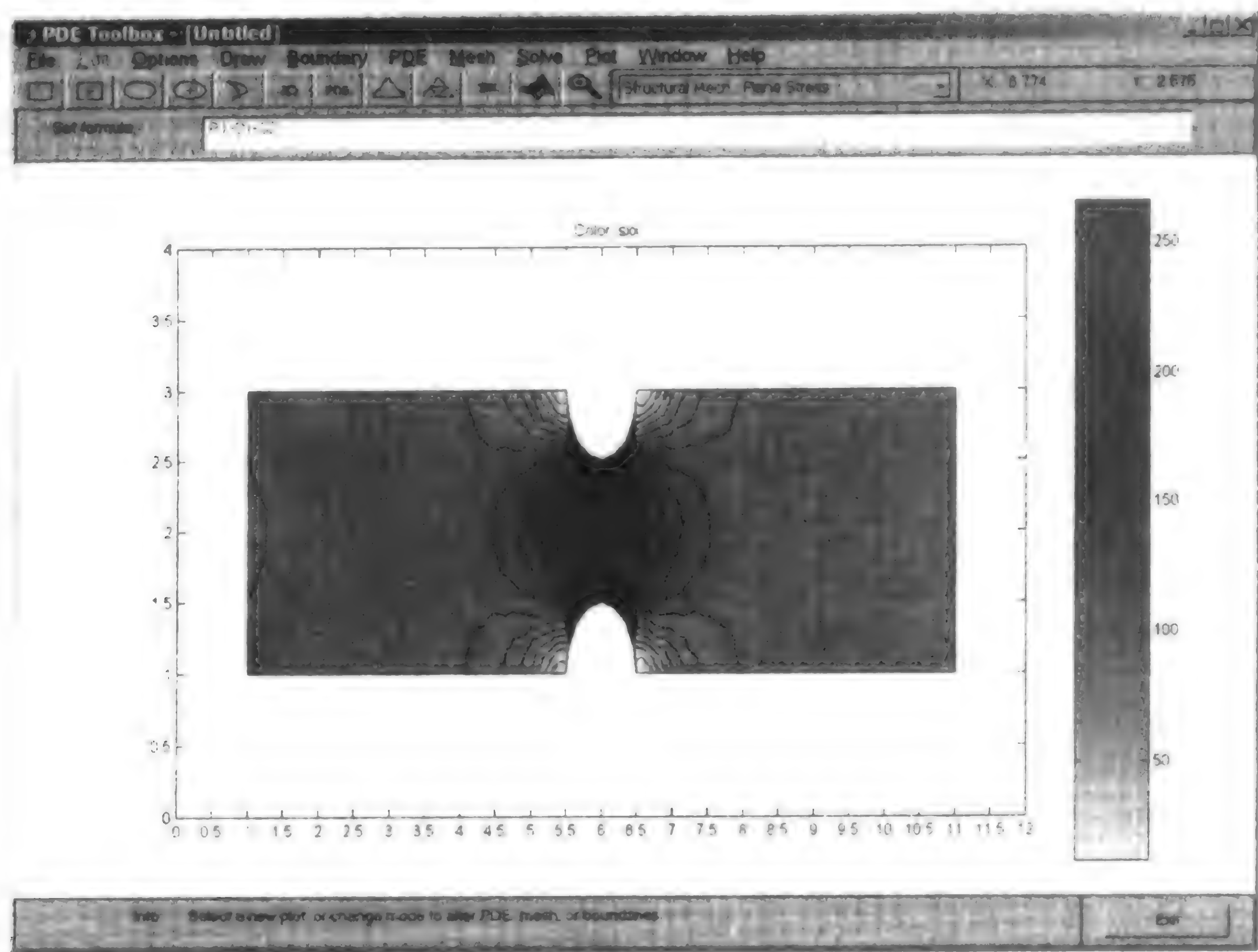


图 8.39 σ_{xx} 应力等高线和色条

图 8.34 至图 8.39 作为与插图封装后的二层压缩文件存储，该压缩文件作为可见图形用于 MS Word 文档中。这些可以按以下方式完成，即在 *File* 菜单上选择 *Print*，在 *Device Option* 数据区域中输入：

```
-deps2  -tiff
```

按 *Enter* 键。在 *Send To* 区域中选择 *File* 然后点击 *Save*。这就打开了一个典型的 *Save As* 窗口，以常用的方式选择路径名和文件名。文件以“.eps”作为扩展名。其他选项参见 print 的帮助文件。

将图形转到 Windows 剪贴板中，使用：

```
-dmeta
```

然后选择 *Printer*。

将由 GUI 生成的数组输出到 MATLAB 命令窗口中

为了确定应力集中系数，需要有应力 σ_{xx} 的最大值。所以，必须将适当的结果传送(输出)到 MATLAB 的命令窗口中去。在这个窗口里使用

```
StressX = pdesmech(p,t,c,u,'tensor','sxx');
```

来获得 σ_{xx} 的数组，然后使用函数 max 寻找最大值。其中，p 给出了网格的坐标点，e 描述了格状三角形的边缘，t 描述角度，u 是计算结果，参见 initmesh 的帮助文件。所有这些数值都从 GUI 输出。在 MATLAB 命令窗口中使用的计算结果必须有一个，可以使用由 MATLAB 提供的、在上面的函数中以参数形式出现的变量名。如果和以前定义的全局变量冲突，则可以更名。同时，必须在 pdesmech 的参数表中更名。

进入 *Mesh* 菜单选择 *Export Mesh* 选项，将 p 和 t 输出到 MATLAB 命令窗口中。这时显示变量名，点击 *OK* 按钮。如果进入 MATLAB 命令窗口键入 p(并回车)，则会出现数组(因为数组可能很大，所以不提倡这样做)。此时可对多个变量更名，然后点击 *OK* 按钮。

进入 *PDE* 菜单并选择 *Export PDE Coefficients* 选项，将参数 c 输出到 MATLAB 命令窗口中去。最后，进入 *Solve* 菜单并选择 *Export Solution* 选项，输出结果数组。

在需要的变量输出到 MATLAB 命令窗口后，用以下两个表达式确定在 x 方向上的最大应力：

```
StressX = pdesmech(p,t,c,u,'tensor','sxx');
MaxSxx = max(StressX)
```

给出 *MaxSxx* = 298.3341(N/cm²)。

为了估计应力集中系数，将一个截面积为 $0.2d$ 的无凹槽平板应力 σ_{xx} 与上面求出的最大应力比较。无凹槽平板的应力 σ_{xu} 近似等于：

$$\sigma_{xu} = \frac{(100)(0.2)D}{0.2d} = 200\ \text{N/cm}^2$$

其中，$D/d = 2$，应力集中系数 S_{cc} 近似等于：

$$S_{cc} = \frac{298.3341}{200} = 1.491$$

这个值无论与实验获得的值 1.37 相比，还是与纽伯(Nueber)图表算出的结果 1.45 相比，都令人满意[①]。

---

① 参见 A.P.Boresi, R.J.Schmidt 和 O.M.Sidebottom, *Advanced Mechanics of Materials*, 第五版, John Wiley & Sons, New York, NY, 1993 年, 第 582～584 页。

注意:在使用 pdetool 来解决其他问题之前,应该使用 clear 来清除变量 p,t,c 和 a,变量使用新名就不必使用 clear。

8.8 应用函数一览

第 8 章引用的函数及其说明如表 8.7 所示。

表 8.7 第 8 章引用的 MATLAB 函数

MATLAB 函数	说明
cross	向量叉乘
dot	向量点乘
norm	向量和矩阵的模
pdetool	为 PDE 工具箱提供 GUI
pdesmech	计算结构力学张力的函数

练习

8.1 将图 8.40 中 F_1,F_2,F_3 及其合力表示为向量。确定结果值和以度数表示的方向余弦角用角度表示。

[答案:$F_1=[56.5685\ \ -42.4264\ \ -70.7107]$;$F_2=[-26.9489\ \ -33.6861\ \ -67.3722]$;$F_3=[28.6401\ 66.8268\ \ -95.4669]$;$R=F_1+F_2+F_3=[58.2597\ \ -9.2857\ \ -233.5497]$,$|R|=240.8856$;$\alpha_R=[76.0039\ 92.2092\ 165.8235]$。]

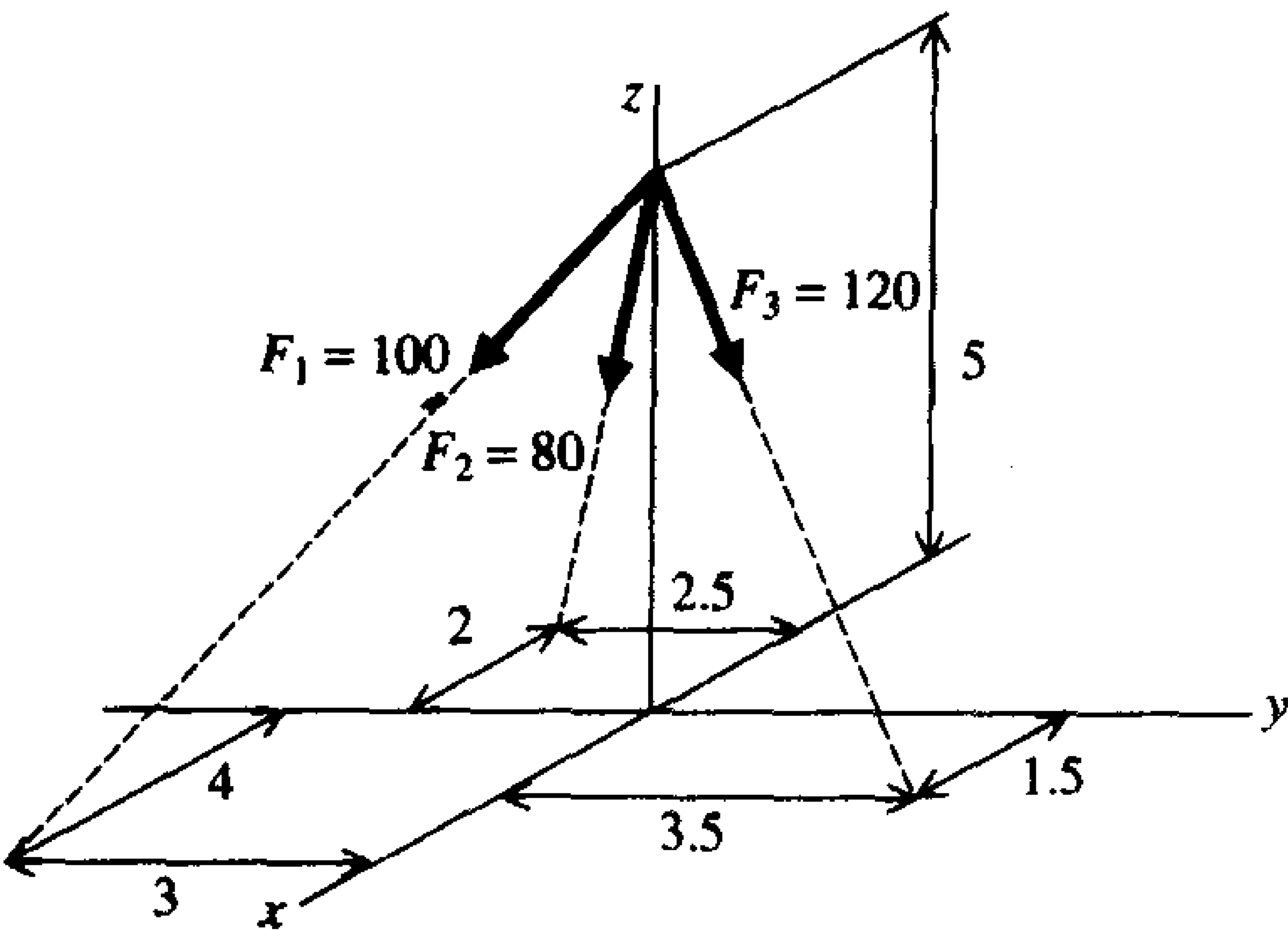


图 8.40 习题 8.1 各力的方向

8.2 绳系固定在图 8.41 所示的坐标中。求力 F_1,F_2 和 F_3 的大小以及各分量的大小。

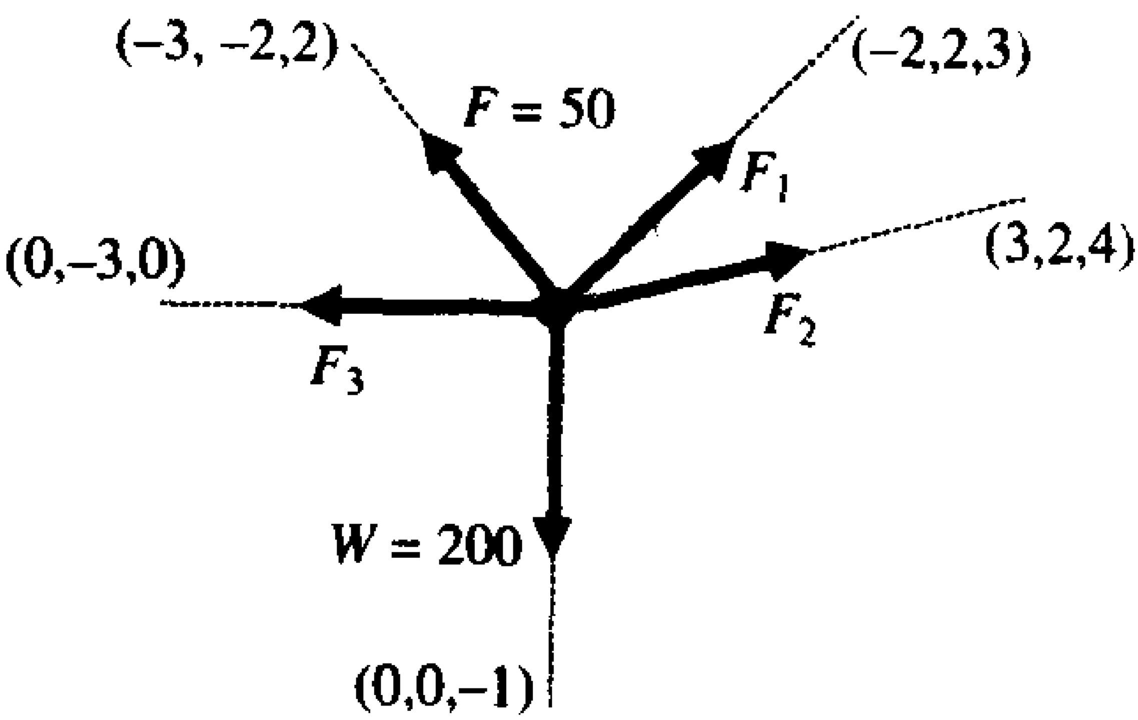


图 8.41 习题 8.2 各力的方向

[答案：$F_1=[-44.9080\ 44.9080\ 67.3620]$；$|F_1|=92.5802$；$F_2=[81.2883\ 54.1992\ 108.3844]$；$|F_2|=145.9170$；$F_3=[0\ -74.8467\ 0]$；$|F_3|=74.8467$。]

8.3 使用 8.2.1 节中的技术，绘出梁的无量纲挠度、转角、弯矩和沿梁的剪力。根据下列边界条件和载荷，给出上述各量的极大值和位置。

(a)在 $\eta=0$ 和 $\eta=1$ 处固定，沿梁有均布载荷。[按此条件用解析法① 求出 $y_{max}(\eta=0.5)=-1/384$，$\theta_{max}(\eta=0.2113$ 或 $\eta=0.7887)=-0.0080$，$M_{max}(\eta=0$ 或 $1)=-1/12$，$V_{max}(\eta=0$ 或 $1)=0.500$。]

(b)在 $\eta=0$ 处简单支撑，在 $\eta=1$ 处固定。沿梁有均布载荷。[按此条件用解析法②得到 $y_{max}(\eta=0.4215)=-1/185$，$\theta_{max}(\eta=0)=-1/48$，$M_{max}(\eta=1)=-1/8$，$V_{max}(\eta=1)=5/8$。]

(c)在 $\eta=0$ 和 $\eta=1$ 处固定。在 $\eta=0$ 处为 1，在 $\eta=1$ 处为 1 时为三角形载荷。[按此条件用解析法③ 得到 $y_{max}(\eta=0.525)=-1/764$，$\theta_{max}(\eta=0.8077)=0.004\,27$，$M_{max}(\eta=1)=-1/20$，$V_{max}(\eta=1)=-7/20$。]

8.4 对于一个承受垂直的静(不变)传输载荷 F_t 的齿轮，接触应力为④：

$$\sigma_c = Z_E\sqrt{\frac{K_v K_H F_t}{d_w b Z_I}} \quad \text{N/mm}^2$$

式中，Z_I 是抗点蚀的几何系数，小齿轮(齿轮 1)的节距直径 $d_w=2R_{P_1}$，R_{P_1} 在表 8.4 中定义，Z_E是弹性系数。其余的值在 8.3 节中定义。

弹性系数的公式如下：

$$\frac{1}{Z_E} = \sqrt{\pi\left[\frac{1-v_1^2}{E_1}+\frac{1-v_2^2}{E_2}\right]}$$

式中，v_1 和 v_2 分别是小齿轮和齿轮的泊松比，E_1 和 E_2 分别是小齿轮和齿轮的弹性模量，当小齿轮和齿轮都是钢材料时，$Z_E=190(\text{N/mm}^2)^{1/2}$。

抗点蚀几何系数为：

$$Z_I = \frac{\rho_1\rho_2\cot(\varphi)}{Cd_p}$$

$$\rho_1 = \sqrt{R_{T_1}^2 - R_{b_1}^2} - m\pi\cos(\varphi_s)$$

$$\rho_2 = (R_{b_1}+R_{b_2})\tan(\varphi) - \rho_1$$

其中，小齿轮的节圆直径 $d_p=N_1 m$，其余数值在表 8.4 和表 8.5 中定义。

要想计算出接触应力 σ_c，必须确定允许应力 σ_H，并验证 σ_H 大于 F_t 引起的接触应力，即 $\sigma_c\leqslant\sigma_H$。允许接触应力如下：

$$\sigma_H = \frac{\sigma_{HP} Z_N Z_w}{F_{sc} Y_Z}$$

---

① 参见 W.Beitz 和 K.H.Kuttner，出处同上，第 B24 页。

② 参见 W.Beitz 和 K.H.Kuttner，出处同上，第 B23 页。

③ 参见 W.Beitz 和 K.H.Kuttner，出处同上，第 B24 页。

④ 参见 J.R.Colbourne，出处同上，1987 年。

下式适用于油和齿轮的温度低于 120℃的情况。F_{sc}值是点蚀的安全系数，Y_Z是8.3 节中定义的可靠系数，Z_N是抗点蚀应力循环系数，σ_{HP}是对于点蚀允许的接触应力数，Z_w是抗点蚀的硬度比例系数。

完全硬钢齿轮的抗点蚀允许的接触应力值由下式估计：

$$\sigma_{HP} = 2.41B_H + 237$$

适用于二级钢。

下式：

$$\sigma_{HP} = 2.22B_H + 200$$

适用于一级钢，布里涅耳硬度范围是 $180 \leqslant B_H \leqslant 400$。

抗点蚀应力循环系数由下式给出：

$$Z_N = 1.4723 \qquad n_L < 10^4$$

$$Z_N = 2.466 n_L^{-0.056} \qquad n_L \geqslant 10^4$$

其中，n_L是单向载荷循环次数。

全硬齿轮驱动的小齿轮表面抗点蚀硬度比系数用下式估算：

$$Z_w = 1 + 0.000\,75 e^{-0.448R_z}(450 - B_{H_2}) \qquad R_z \leqslant 1.6$$

$$Z_w = 1 \qquad R_z > 1.6$$

其中，R_z 是小齿轮的表面磨光度，单位是 μm；B_{H_2} 是齿轮的布里涅耳硬度，范围是 $180 \leqslant B_{H_2} \leqslant 400$。

考虑下面一对钢齿轮：

$m = 10$ mm	$b = 45$ mm	$n = 1800$ r/m	$B_{H_2} = 260$
$N_1 = 28$	$\varphi_s = 20°$	$T = 2500$ Nm	
$N_2 = 75$	$C = 525$ mm	$R_{T_1} = 153.9$ mm	

根据下列条件确定接触应力和允许接触应力：

安全系数是 1.2
单向载荷循环次数是 4×10^8
用 2 级钢
小齿轮的表面磨光度是 1.1 μm
失效率希望低于 1/100
$Q_v = 8$

该齿轮是表 8.3 中定义的二型齿轮。

[答案：$\sigma_c = 857.2436$ N/mm², $\sigma_H = 636.2982$ N/mm²。]

8.5 用例 8.21 中的结果绘出连杆的三个位置点，仿照图 8.25 标出点 P_1 的路径。用 fsolve 通过下式计算 Ψ 和 γ 值，从而确定 P_1 的路径，对 $\phi(\theta \leqslant \phi \leqslant \theta + \phi_2)$16 等分：

$$\mathbf{Z}_2(e^{j\phi} - 1) - \mathbf{Z}_4(e^{j\Psi} - 1) + (\mathbf{Z}_5 - \mathbf{Z}_6)(e^{j\gamma} - 1) = 0$$

设上式中的实部和虚部均为 0，可得到 fsolve 所需的两个方程，对方程不使用代数的方法而使用数值计算的方法。

8.6 用于振动的平面盘形从动件如图 8.42 所示，在(x,y)坐标系中，凸轮轮廓的坐标 R_x 和 R_y，刀具坐标 C_x 和 C_y 有以下关系①，分别为：

$$R_x = R\cos(\theta + \varphi + \xi_0) \qquad R_y = R\sin(\theta + \varphi + \xi_0)$$

$$C_x = C\cos(\theta + \varphi + \xi_0 - \gamma_0) \quad C_y = C\sin(\theta + \varphi + \xi_0 - \gamma_0)$$

其中，

$$R = \frac{f + m\sin(\xi)}{\cos(\theta)} \qquad \theta = \arctan\left[\left(\frac{\mathrm{d}\xi/\mathrm{d}\varphi}{1 - \mathrm{d}\xi/\mathrm{d}\varphi}\right)\frac{m\cos(\xi)}{f + m\sin(\xi)}\right]$$

$$C = \sqrt{c_x^2 + c_y^2} \qquad c_x = r_c\sin(\theta) \qquad c_y = R + r_c\cos(\theta)$$

$$\gamma_0 = \arctan\left(\frac{c_y}{c_x}\right) \qquad \xi_0 = \arcsin\left(\frac{r_b - f}{m}\right) \qquad \xi = \xi(\varphi) = \xi_0 + s(\varphi)/R_0$$

对于点 R_0的摆线运动，$s(\varphi)$在 8.5.1 节中给出。当 $\beta = 60°$，$r_b = 3.25$，$h = 0.5$，$r_c = 0.5$，$f = 0.5$，$m = 5$，$R_0 = 9$ 时，画出一个与图 8.29 和图 8.31 相似的图形。

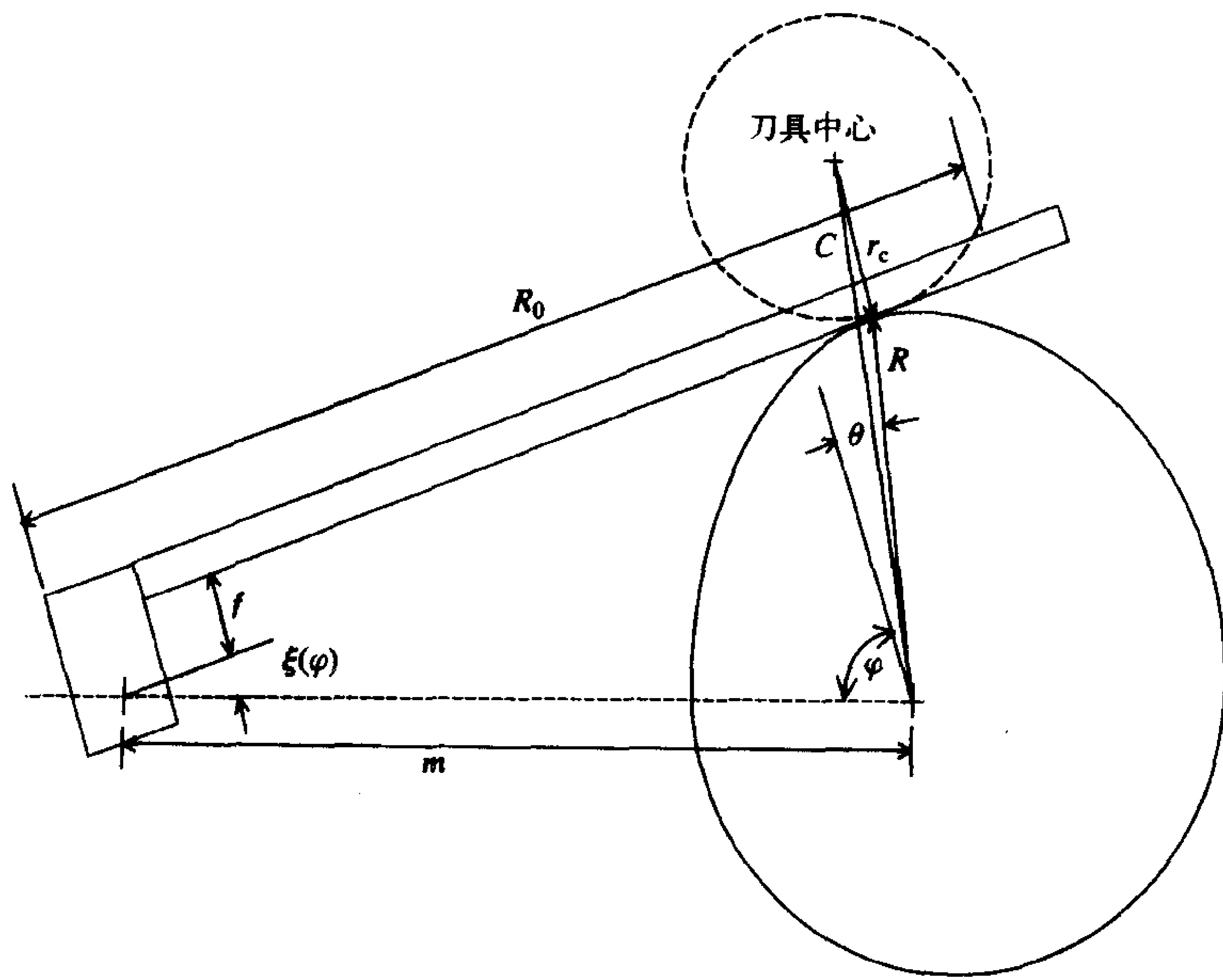


图 8.42 振动平面从动件

8.7 外部增压中心凹槽推进空气轴承如图 8.43 所示。使用表 8.7 的定义，给定 P_s，通过轴承的正常物质流由下式给出②：

$$m' = P_s\sqrt{\left(\frac{P_1}{P_s}\right)^{2/k} - \left(\frac{P_1}{P_s}\right)^{(k+1)/k}}$$

其中，$m' = m/m_o$，$P_1 = p_1/p_a$，$P_s = p_s/p_a$，P_1的解为：

$$P_1 = \sqrt{1 + BP_s^{(k-1)/2k}P_1^{1/k}\sqrt{P_s^{(k-1)/k} - P_1^{(k-1)/k}}}$$

---

① 参见 A.G.Erdman 和 G.N.Sandor，出处同上，第 387～389 页。

② 参见 W.A.Gross, *Gas Film Lubrication*, John Wiley & Sons, New York, NY, 1962 年，第 5 章。

当 $P_1/P_s = p_1/p_s > p_c = [2/(k+1)]^{k/(k-1)}$ 时，P_1 的解为：

$$P_1 = \sqrt{1 + BP_s P_c^{1/k} \sqrt{1 - P_c^{(k-1)/k}}}$$

当 $P_1/P_s \leq P_c$ 时，是数据流拥塞的区域[马赫(Mach)数等于 1]。

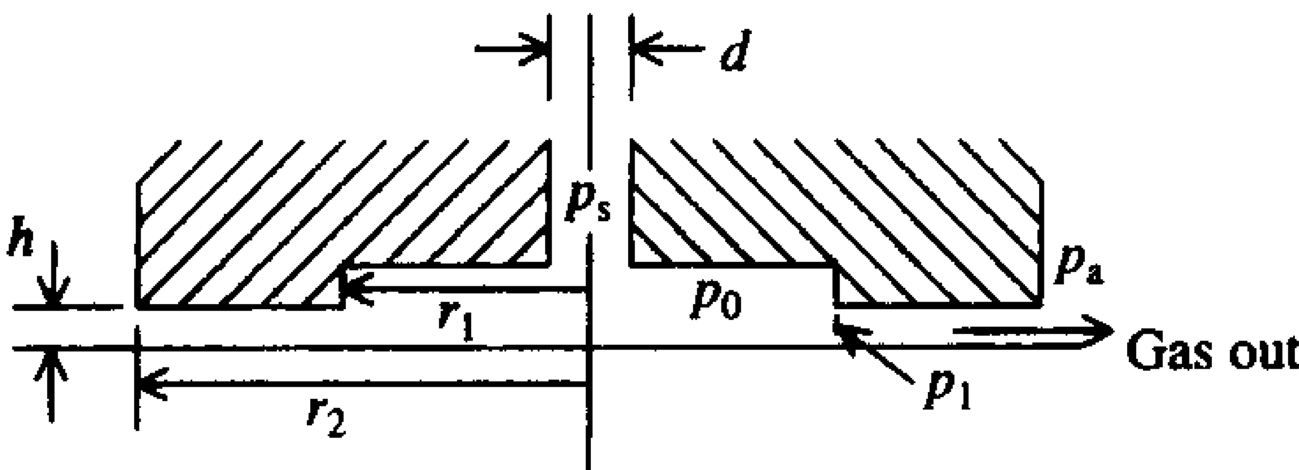


图 8.43　空气轴承的几何尺寸

表 8.8　空气轴承公式中量的定义

量	定义
C_D	泄露系数
$A_o = \pi d^2/4$	限制域
P_s	提供压力
P_o	凹口压力
p_1	入口压力
p_a	周围压力
h	气膜厚度
μ	气体粘度
k	热比(空气为 1.4)
R	气体常数(通用气体常数除以分子量)
θ	温度(绝对温度)
$B = \frac{12 C_D \mu A_o}{h^3 p_a \pi} \ln\left(\frac{r_2}{r_1}\right) \sqrt{\frac{(k-1)R\theta}{2k}}$	轴承参数
$m_o = \frac{1}{C_D A_o p_a} \sqrt{\frac{(k-1)R\theta}{2k}}$	质量流参数
m	质量流
W	轴承载荷

轴承通常能够承受的载荷为：

$$W' = \frac{\sqrt{\pi}}{2} \frac{r_1^2}{r_2^2} A_2 \exp(P_1^2/A_2^2)\left[\operatorname{erf}\left(\frac{P_1}{A_2}\right) - \operatorname{erf}\left(\frac{1}{A_2}\right)\right]$$

其中，$W' = W/(\pi r_2^2 p_a)$，erf 是误差函数，并且：

$$A_2 = \sqrt{\frac{(1 - P_1{}^2)}{2\ln(r_1/r_2)}}$$

(a)当 $B = 2$ 时，分别确定(i) $P_s = 2$ 和(ii) $P_s = 10$ 时的 P_1 值。在使用函数 fzero 时，设置的搜索范围为[0.1 P_s]。

[答案:(i) $P_1 = 1.4016$,(ii)$P_1 = 2.4852$。]

(b)当 $r_1/r_2 = 0.1$,$B = 2$ 时,对应(i)$P_s = 2$ 和(ii)$P_s = 10$ 时,确定 m' 和 W' 的值。

[答案:(i)$m' = 0.4822$, $W' = 0.0954$;(ii)$m' = 2.5880$, $W' = 0.4147$。]

8.8 使用 8.7 节介绍的过程确定有圆形孔或椭圆形孔平板上的最大应力。设有三种情况如图 8.44 所示。假定孔的边界无应力。对于这三种情况,令 $\sigma = 1$,表示 y 轴的正方向;令 $\sigma = -1$,表示 y 轴的负方向。再令弹性模量等于 200×10^3,使用一次 *Refine Mesh* 选项。对于情况(a),令 $a = 1$(单位);对于情况(b)和(c),令 $b = 1$(单位)和 $a = 2$。

[答案:

情况(a):对于圆形孔的情况,用解析法获得的最大应力比[①] 是 $\sigma_{yy}/\sigma = 3$;用 PDE 方法给出 2.9122,或低于 2.9%。

情况(b):对于椭圆形孔且载荷垂直于椭圆长轴的情况,用解析法获得[②] 的结果是:$\sigma_{yy}/\sigma = 1 + 2a/b$,数值 $\sigma_{yy}/\sigma = 5$。用 PDE 方法给出的是 $\sigma_{yy}/\sigma = 4.2264$,低于 15.5%。

情况(c):对于椭圆形孔且载荷垂直于椭圆短轴的情况,用解析法获得[③] 结果是:$\sigma_{yy}/\sigma = 1 + 2b/a$,数值 $\sigma_{yy}/\sigma = 2$。用 PDE 方法给出的是 $\sigma_{yy}/\sigma = 2.3552$,高出 17.8%。]

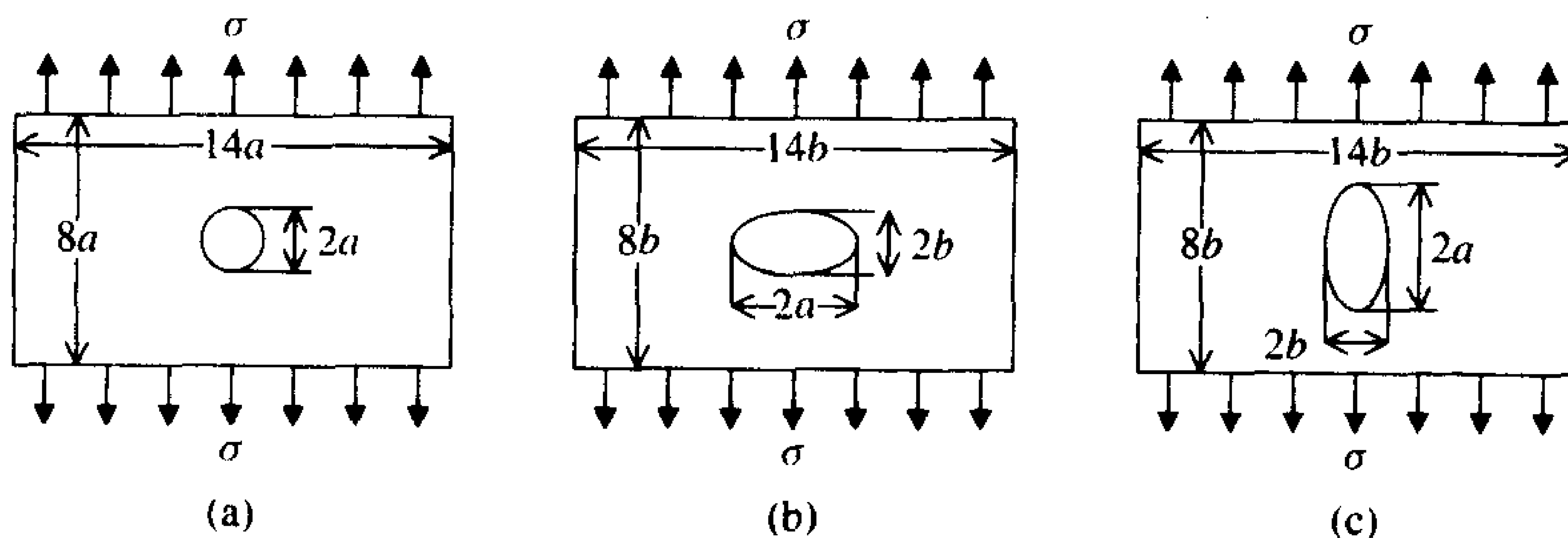


图 8.44　各种平板(a) 有圆孔的平板;(b) 有椭圆孔的平板且载荷垂直于椭圆长轴;(c) 有椭圆孔的平板且载荷垂直于椭圆短轴

8.9 考虑如图 8.45 所示的一个贝勒维尔(Belleville)圆锥弹簧。无量纲载荷 P',无量纲弹簧比(弹性常数)k' 和无量纲最大压缩应力 σ',均作为弹性形变 δ 的函数[④],每个量的表达式如下:

$$P' = \frac{P}{P_0} = C_1 C_2 \qquad P_0 = \frac{Et^4}{(1-\nu^2)b^2}$$

$$k' = \frac{k}{k_0} = C_2[1 + 1.5d_t^2 - 3d_t h_t + h_t^2] \qquad k_0 = \frac{P_0}{t}$$

① 参见 Boresi 等,出处同上,第 566 ~ 567,569 页。

② 参见 Boresi 等,出处同上,第 568 ~ 569 页。

③ 参见 Boresi 等,出处同上,第 570 页。

④ 参见 A. H. Burr 和 J. B. Cheatham, *Mechanical Analysis and Design*,第二版,Prentice Hall, Upper Sanddle River, NJ, 1995 年,第 652 ~ 656 页。

$$\sigma'_{max} = \frac{\sigma'_{max}}{\sigma_0} = -d_t[C_3(h_t - 0.5d_t) + C_4] \qquad \sigma_0 = \frac{P_0}{t^2}$$

其中,

$$C_1 = 0.5d_t^3 - 1.5h_td_t^2 + (1 + h_t^2)d_t$$

$$C_2 = \pi\left(\frac{\alpha + 1}{\alpha - 1} - \frac{2}{\ln(\alpha)}\right)\left(\frac{\alpha}{\alpha - 1}\right)^2$$

$$C_3 = \frac{\alpha^2}{(\alpha - 1)^2}\left(\frac{\alpha - 1}{\ln(\alpha)} - 1\right)$$

$$C_4 = \frac{\alpha^2}{2(\alpha - 1)}$$

并且,ν 是泊松比,E 是弹性模量,$\alpha = b/a$,$h_t = h/t$,$d_t = \delta/t$。最大压缩应力发生在圆锥的上表面内边界 a 处,并且通常规定 $\sigma_{max} = \sigma_p/F_s$。其中,$\sigma_p$ 是最大允许应力,F_s 是安全系数。

当 $d_t = 0.6667h_t$,$P' = 2.0$,$\sigma'_{max} = -4.5$ 时,求 α 和 h_t 的值,以及 k' 的值。

[答案:$\alpha = 3.3049$, $h_t = 1.5533$, $k' = 0.2460$。]

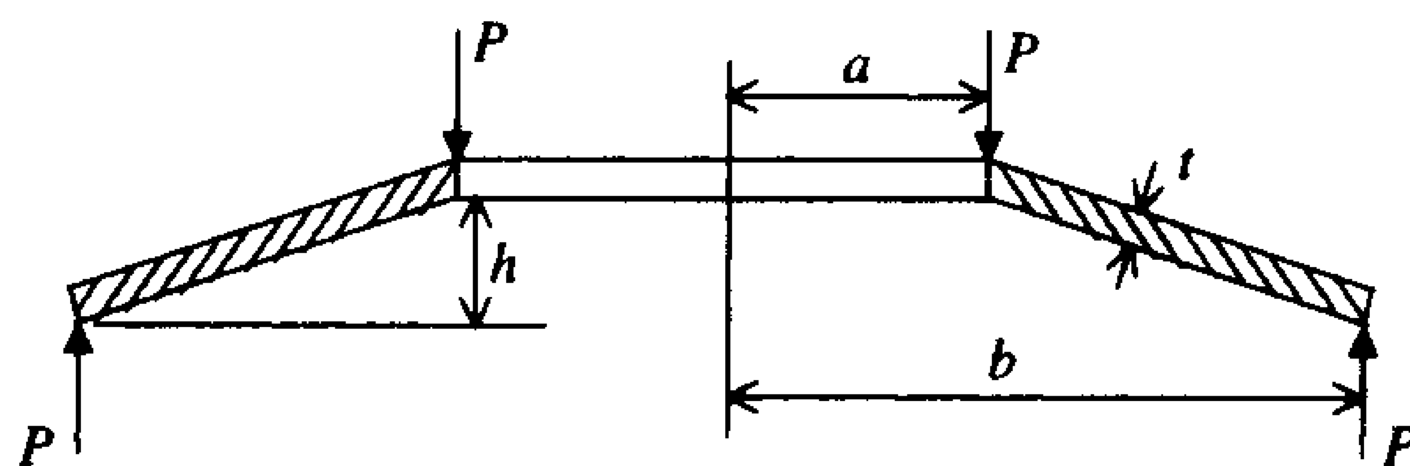


图 8.45 贝勒维尔圆锥弹簧的横截面

8.10 参见图 8.46,承受周期载荷的螺旋压缩弹簧的尺寸由以下方程决定。第一个是安全系数 F_s,取决于弹簧圈中的应力。

$$F_s = \frac{S_{eL}(S_u - \sigma_L)}{S_{eL}(\sigma_m - \sigma_L) + S_u\sigma_a} > 1$$

其中,

$$\sigma_L = K_sF_{min}K_1 \qquad K_1 = \frac{8C}{\pi d^2} \qquad C = \frac{D}{d}$$

$$\sigma_m = K_sF_mK_1 \qquad F_m = 0.5(F_{max} + F_{min})$$

$$\sigma_a = K_wF_aK_1 \qquad F_a = 0.5(F_{max} - F_{min})$$

$$K_s = 1 + \frac{1}{2C} \qquad K_w = \frac{4C - 1}{4C - 4} + \frac{0.615}{C}$$

$$S_u = 0.67Ad^b \qquad S_e = \frac{0.707S_{eL}S_u}{S_u - 0.707S_{eL}}$$

F_{max}和 F_{min}分别是作用在弹簧上的最大和最小载荷。S_{el}是无限寿命疲劳极限,S_e是反向载荷的疲劳极限,S_u是极限剪切强度,D 是弹簧的平均直径,d 是弹簧钢线的直径。对于冷拔钢丝,常量 $a = 141\ 040$ psi,常量 $b = -0.1822$。对于硬化弹簧钢,$d < 0.4$ in,$S_{eL} = 67\ 500$ psi。

第二个方程确定弹簧比(常数)k,由下式给出:

$$k = \frac{dG}{8N_a C^3}$$

式中,G 是这种材料的剪切弹性模量,N_a 是弹簧的匝数。对于钢 $G = 11.5 \times 10^6$psi。最后的方程确定弹簧的自然长度 L_f,弹簧圈的直径 d,使得钢弹簧不弯曲的最大负荷 F_{max}。两端连接时,有下式:

$$\frac{F_{max}}{kL_f} = 0.8125\left[1 - \sqrt{1 - 6.865(Cd/L_f)^2}\right]$$

式中,$Cd = D$,并且:

$$L_f = d(N_a + 2) + 1.15F_{max}/k - 0.15F_{min}/k$$

提供 15%的冲撞允许范围。

对于 $N_a = 7$, $F_{max} = 200$ lb, $F_{min} = 40$ lb, $F_s = 1.4$ 和 $k = 125$ lb/in 的硬化弹簧钢,求出 D 和 d,问弹簧是否会发生弯曲?

[答案:$d = 0.251$ in, $D = 1.8682$ in, $C = 7.4432$。弹簧没有发生变形。]

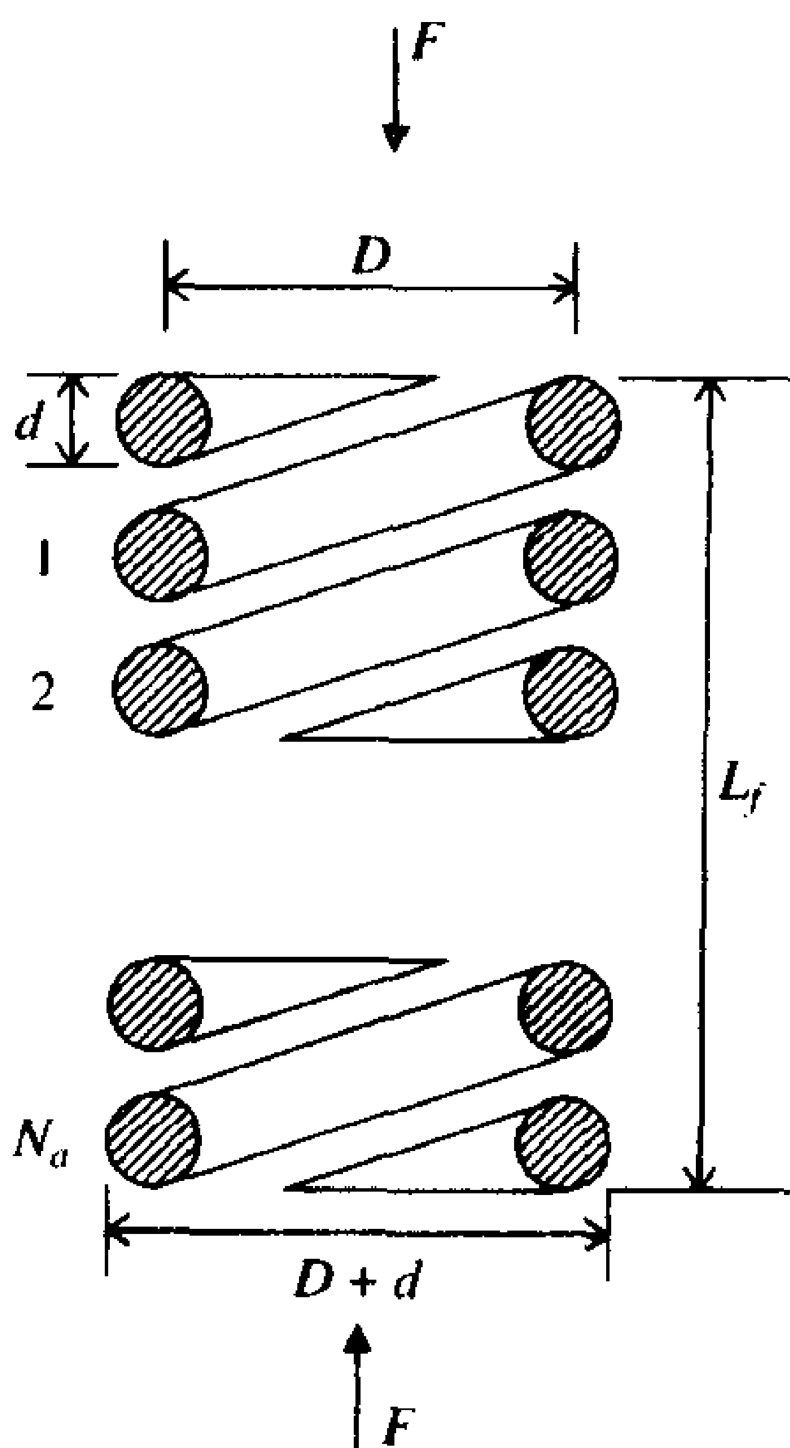


图 8.46　关于螺旋弹簧的术语

8.11　驱动螺杆升起载荷 W 所需的转矩为:

$$T_{raise} = \frac{Wd_p}{2}\left[\frac{\mu + \tan(\lambda)\cos(\alpha)}{\cos(\alpha) - \mu\tan(\lambda)} + \mu_c \frac{d_c}{d_p}\right]$$

式中,λ 是超前角,由下式求出:

$$\tan(\lambda) = \frac{L}{\pi d_p}$$

μ 是螺纹的摩擦系数,μ_c 是环管(螺母)的摩擦系数,d_p 是螺纹的节距直径,d_c 是螺母的平均半径,$L = mp$ 是螺距,$p = 1/N_t$ 是节距,N_t 是每英寸螺纹的头数,m 是起始螺纹的头数。对于 Acme 螺纹来说,角度 α 为 14.5°,$1.9° < \lambda < 6°$,螺纹和螺母的效率 e 为:

$$e = \frac{WL}{2\pi T_{raise}} = \tan(\lambda)\left[\frac{\mu + \tan(\lambda)\cos(\alpha)}{\cos(\alpha) - \mu\tan(\lambda)} + \mu_c \frac{d_c}{d_p}\right]^{-1}$$

当 $\mu_c = 0$ 时(参见习题 6.3),e 由下式表示:

$$e = \frac{\cos(\alpha) - \mu\tan(\lambda)}{\cos(\alpha) - \mu\cot(\lambda)}$$

以每分钟 v_r 英尺的速度升起载荷需要的马力为:

$$hp = \frac{v_r T_{raise}}{2626 d_p}$$

式中,T_{raise} 的单位是 in-lb。

对于截距直径为 1.0 in,超前角为 4°的一条单 Acme 线($m = 1$),当环管的平均直径为 1.25 in 时,求以每分钟 15 ft 的速率升高 800 lb 所需的转矩和马力。线的摩擦系数为 0.13,环管的摩擦系数为 0.04。问这个系统的效率是多少?

[答案:$T_{raise} = 102.4558$ in-lb, $e = 27.3003\%$, $hp = 0.585\,024$。]

8.12 在一个扁节链中,连接板的疲劳强度决定功率极限为:

$$hp = K_s N_1^{1.08} n_1^{0.9} p^{3-0.07/p}$$

式中,N_1 是较小链轮的齿数,n_1 是较小链轮的转速,单位是 r/m。p 是截距,也是链栓中心之间的距离。对于常规重量链条 $K_s = 0.004$。当 $N_1 = 21$ 齿,$n_1 = 1750$ r/m,并且 $hp = 10$ 时,求节距。

[答案:$p = 0.464\,53$。]

参考文献

1. "Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth," AGMA Standard ANSI/AGMA 2002-C95, American Gear Manufacturers Association, 1500 King Street, Alexandria, VA, 22314.
2. W. Beitz and K. H. Kuttner, Eds., *Handbook of Mechanical Engineering*, Springer-Verlag, New York, NY, 1994.
3. S. R. Bhonsle and L. J. Weinmann, *Mathematical Modeling for the Design of Machine Components*, Prentice Hall, Upper Saddle River, NJ, 1999.
4. A. H. Burr and J. B. Cheatham, *Mechanical Analysis and Design*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1995.
5. J. R. Colbourne, *The Geometry of Involute Gears*, Springer-Verlag, New York, NY, 1987.
6. A. D. Dimarogonas, *Computer-Aided Machine Design*, Prentice Hall, Upper Saddle River, NJ, 1989.
7. A. G. Erdman and G. N. Sandor, *Mechanical Design: Analysis and Synthesis*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1991.
8. R. C. Hibbeler, *Engineering Mechanics: Statics*, 8th ed., Prentice Hall, Upper Saddle River, NJ, 1998.
9. P. J. Jensen, *Cam Design and Manufacture*, 2nd ed., Marcel Dekker, Inc., New York, NY,

1987.

10. R.L.Norton, *Design of Machinery*, McGraw-Hill, New York, NY, 1992.
11. R.L.Norton, *Machine Design: An Integrated Approach*, Prentice Hall, Upper Saddle River, NJ, 1996.
12. H.A.Rothbart, *Cams: Design, Dynamics, and Accuracy*, John Wiley & Sons, New York, NY, 1956.
13. J.E.Shigley and C.R.Mischke, *Mechanical Engineering Design*, 5th ed., McGraw-Hill, New York, NY, 1989.
14. A.H.Slocum, *Precision Machine Design*, Prentice Hall, Upper Saddle River, NJ, 1992.
15. M.F.Spotts and T.E.Shoup, *Design of Machine Elements*, Prentice Hall, Upper Saddle River, NJ, 1998.
16. C.E.Wilson, *Computer Integrated Machine Design*, Prentice Hall, Upper Saddle River, NJ, 1997.

第9章 动力学与振动

本章将介绍几种不同的方法,分别用于分析刚体动力学、单自由度和多自由度线性与非线性系统中的自由振动和强迫振动以及梁的振动。

9.1 轨迹

如图9.1所示,重力场中有两个物体,其中质量为 m_2 的物体固定,而质量为 m_1 的物体绕 m_2 做平面圆周运动。做圆周运动的 m_1 物体的轨道半径用变量 r 表示,角度用变量 θ 表示。

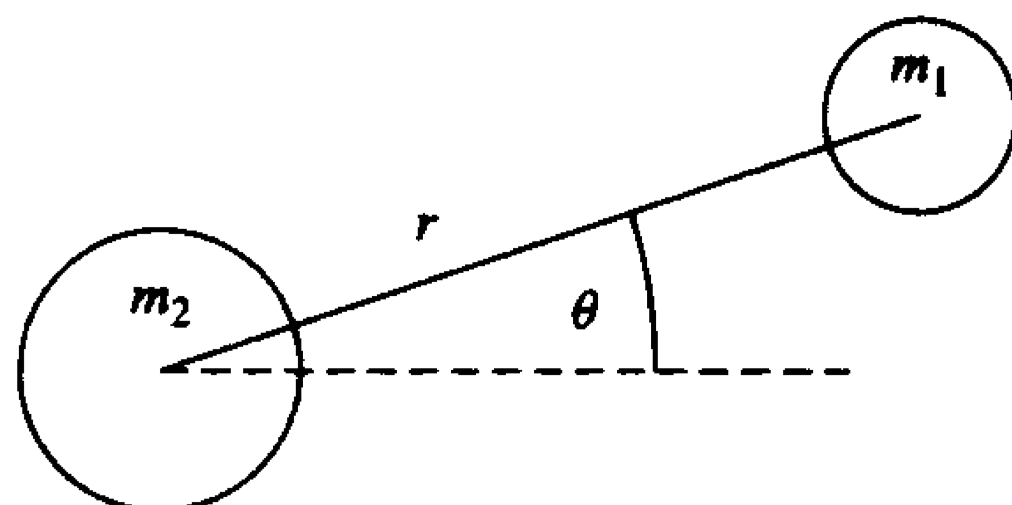


图9.1 两物体系统

卫星绕地球转动时,m_2 等于地球的质量,m_1 等于卫星的质量,r 为卫星球心与地球球心间的距离。其运动轨迹由下列方程组决定[①][②]:

$$\frac{\mathrm{d}^2 r}{\mathrm{d}\tau^2} - r\left(\frac{\mathrm{d}\theta}{\mathrm{d}\tau}\right)^2 = -\frac{4\pi^2}{r^2} \tag{9.1}$$

$$r\frac{\mathrm{d}^2\theta}{\mathrm{d}\tau^2} + 2\frac{\mathrm{d}r}{\mathrm{d}\tau}\frac{\mathrm{d}\theta}{\mathrm{d}\tau} = 0$$

式(9.1)中,$\tau = t/P_c$。其中,t 是时间变量,P_c 为物体在地球表面做圆周运动的周期。在地球表面,r 约等于地球半径,即 $r = 6.373 \times 10^6$ m。

尽管由式(9.1)可得出非线性系统的一个封闭解[③],但仍需求出微分方程在给定初始条件下的数值解。初始条件为:

$$r(0) \quad \mathrm{d}r(0)/\mathrm{d}\tau \quad \theta(0) \quad \mathrm{d}\theta(0)/\mathrm{d}\tau$$

有时可简写为:

$$(r(0), \mathrm{d}r(0)/\mathrm{d}\tau, \theta(0), \mathrm{d}\theta(0)/\mathrm{d}\tau)$$

式(9.1)给出了动态微分系统的一个例子,描述了状态变量 r 和 θ 随 τ 的变化。一般来说,动态微分系统由一组微分方程构成,描述所研究的状态随独立变量(比如时间)的变化而变化的情况。正如式(9.1)所示,当一个动态系统由一组常微分方程构成时,可以用对独立变量的数值积分来求出该公式在给定初始条件下的解。

---

① 参见 D.T. Greenwood, *Principles of Dynamics*,第二版, Prentice Hall, Englewood Cliffs, NJ, 1988年,第5章。

② 参见 F.C. Moon, *Applied Dynamics with Applications to Multibody and Mechatronic Systems*, John Wiley & Sons, New York, NY, 1998年,第7章。

③ 参见 D.T. Greenwood,出处同上。

例9.1 不同条件下的圆周运动

下面求质量为 m_1 的物体在表9.1所示的三组初始条件下的圆周运动。尽管不能提前推断每种情况下实现的运动轨迹类型,但经过对数值结果分析可得到运动轨迹类型,并显示在输出表格的最后一列。

引入新状态变量:

$$x_1 = r \qquad x_3 = \theta$$

$$x_2 = \frac{dr}{d\tau} \qquad x_4 = \frac{d\theta}{d\tau}$$

重写式(9.1)可得四个一阶微分方程:

$$\begin{aligned}
\frac{dx_1}{d\tau} &= x_2 \\
\frac{dx_2}{d\tau} &= x_1 x_4^2 - \frac{4\pi^2}{x_1^2} \\
\frac{dx_3}{d\tau} &= x_4 \\
\frac{dx_3}{d\tau} &= -\frac{2x_2 x_4}{x_1}
\end{aligned} \tag{9.2}$$

针对三组初始条件求出三种运动轨迹并画出轨迹图形,程序如下:

```
function OrbitMotion
initcond = [2.0,0.0,0.0,1.5;1.0,0.0,0.0,2.0 * pi;2.0,0.0,0.0,4.0];
tspan = linspace(0,5,1000);
options = odeset('RelTol',1e-6,'AbsTol',[1e-6 1e-6 1e-6 1e-6]);
for i = 1:3
  [t,x] = ode45(@orbit,tspan,[initcond(i,:)]',options);
  polar(x(:,3),x(:,1),'k-');
  hold on
end
text(0.50,-1.30,'Elliptical orbit');
text(-2.10,1.00,'Circular orbit');
text(1.75,2.00,'Hyperbolic orbit');

function xdot = orbit(t,x)
xdot = [x(2);x(1) * x(4)^2 - 4.0 * pi^2/x(1)^2;x(4); - 2.0 * x(2) * x(4)/x(1)];
```

数组 *initcond* 中的元素为三组初始条件的值,linspace 函数定义了确定时间内可用1000个等间隔恒定值以保证数值分析过程中尽量不产生无效解。另外,odeset 函数设置相对误差精度等于 10^{-6},还设置了四种状态的绝对误差精度——用 x_j 表示,$j = 1,2,\cdots,4$。

表9.1 3组初始条件

组	$x_1(0)$	$x_2(0)$	$x_3(0)$	$x_4(0)$	轨迹类型
1	2.0	0.0	0.0	1.5	椭圆
2	1.0	0.0	0.0	2π	圆
3	2.0	0.0	0.0	4.0	双曲线

图 9.2 是以上程序执行的结果。由第一种初始条件得到的轨迹为椭圆,由第二种初始条件得到的轨迹为圆,由第三种初始条件得到的轨迹为双曲线,即一非闭合轨迹,表示物体的运动为无限运动①。在这种情况下卫星已脱离绕地球转动的轨道,而不再是一般的卫星运动了。另外,在 $r<1$ 时卫星将摔落到地球表面,因此椭圆轨迹是不符合实际情况的。

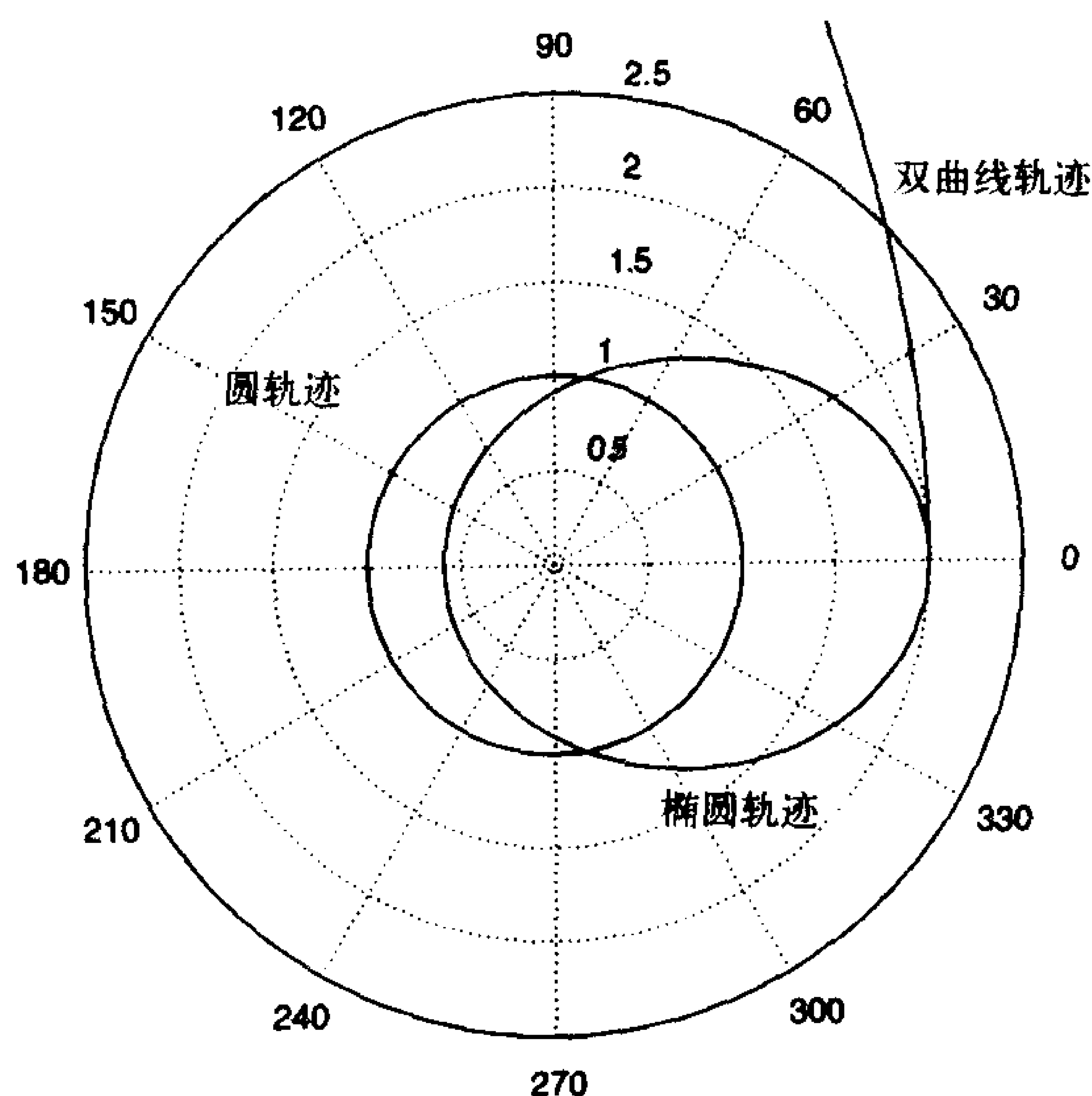


图 9.2 根据表 9.1 给定的初始条件由式(9.1)推出的三种轨迹

用数值积分方法分析类似式(9.1)描述的系统,积分过程中可能会产生无效解。一般系统中有一个状态量不随时间变化[例如式(9.1)中单位质量 $r^2\mathrm{d}\theta/\mathrm{d}\tau$ 的角动量],当产生无效解时,该常量也将随时间变化。无阻尼系统进行数值分析时可根据系统的动能和势能之和是否保持不变来判断是否产生无效解。因此较好的方法是通过改变步长或改变误差精度,或两者同时改变来确定结果是否不同(参考习题 9.1)。

9.2 单自由度系统

9.2.1 概述

如图 9.3 所示,在弹簧 - 质量 - 阻尼器系统中,振体质量为 m,弹簧的线性系数(倔强系数)为 k,非线性系数为 α,阻尼器的阻尼系数为 c。振体受一外力 $F(t)=X_0kf(t)$作用。该方程是研究力学系统的一种典型模型,研究范围可以包括从洗衣机到车辆的振动问题②。

若用 x 表示系统的位移,则运动方程具有下列形式:

$$\frac{\mathrm{d}^2x}{\mathrm{d}\tau^2}+2\zeta\frac{\mathrm{d}x}{\mathrm{d}\tau}+x+\hat{\alpha}x^3=X_of(\tau) \tag{9.3}$$

① 参见 D.T.Greenwood,出处同上,第 211 页。

② 参见 D.J.Inman, *Engineering Vibration*, Prentice Hall, Englewood Cliffs, NJ, 1994; S.S.Rao, *Mechanical Vibrations*,第三版, Addison-Wesley, Reading, MA, 1995 年; B.H.Tongue, *Principles of Vibration*, Oxford University Press, New York, NY, 1996 年。

式中，$\tau=\omega_n t$ 为无量纲时间，阻尼因子 ζ 由下式求出：

$$\zeta=\frac{c}{2m\omega_n} \tag{9.4}$$

式(9.4)中的固有频率 ω_n 为：

$$\omega_n=\sqrt{k/m} \tag{9.5}$$

非线性系数 $\hat{\alpha}$ 由下式确定：

$$\hat{\alpha}=\frac{\alpha}{m\omega_n^2} \tag{9.6}$$

在式(9.3)给出的非线性系统中，当 $\hat{\alpha}$ 为负值时弹簧为软弹簧，当 $\hat{\alpha}$ 为正值时弹簧为硬弹簧①。

引入新变量或状态量：

$$x_1=x$$

$$x_2=\frac{\mathrm{d}x}{\mathrm{d}\tau}$$

式(9.3)转化为状态空间方程形式：

$$\begin{aligned}\frac{\mathrm{d}x_1}{\mathrm{d}\tau}&=x_2\\ \frac{\mathrm{d}x_2}{\mathrm{d}\tau}&=-2\zeta x_2-x_1-\hat{\alpha}x_1^3+X_\mathrm{o}f(\tau)\end{aligned} \tag{9.7}$$

在分析式(9.7)的几种解之前，先创建函数 **FreeOscillation**，以由函数 ode45 调用：

```
function xdot = FreeOscillation(t, x, zeta, AlphaHat)
xdot = [x(2); - 2 * zeta * x(2) - x(1) - AlphaHat * x(1)^3];
```

对于线性系统，有 $AlphaHat=\hat{\alpha}=0$。

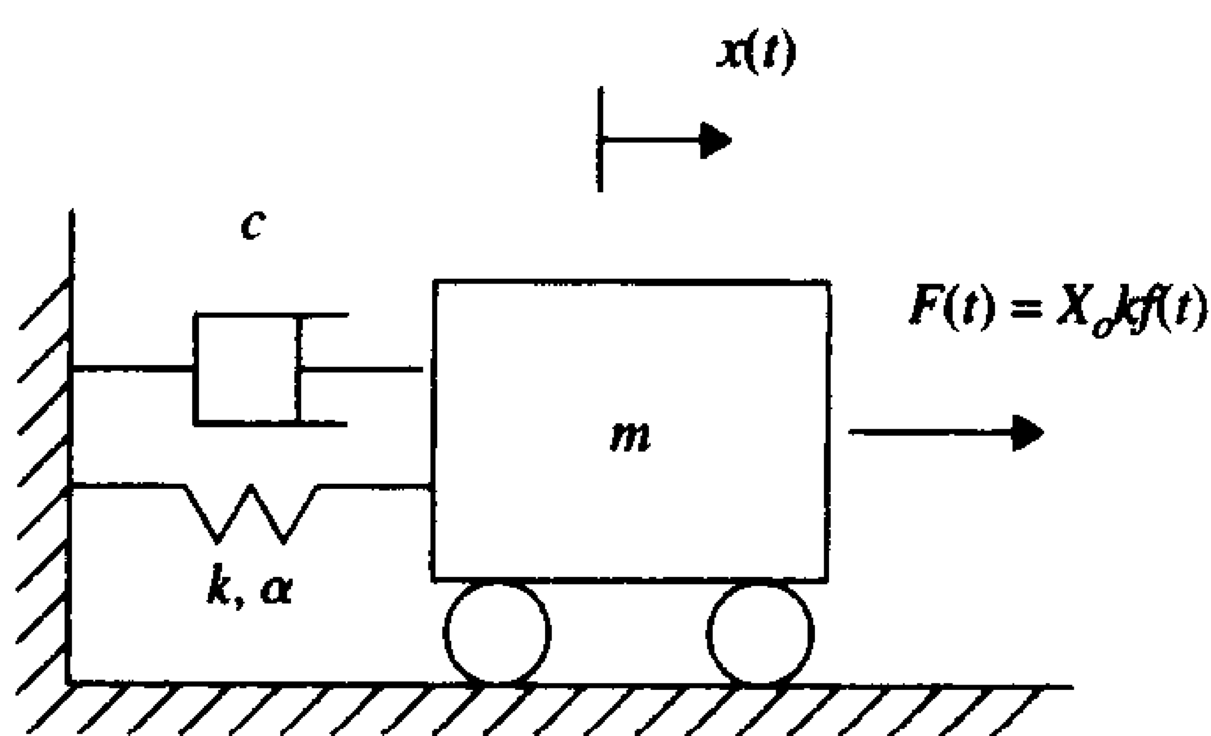


图 9.3　弹簧－质量－阻尼器系统

9.2.2　线性系统的自由振动

当 $\hat{\alpha}=F(\tau)=0$ 时，得到一线性系统的自由振动方程，式(9.3)变为：

$$\frac{\mathrm{d}^2x}{\mathrm{d}\tau^2}+2\zeta\frac{\mathrm{d}x}{\mathrm{d}\tau}+x=0 \tag{9.8}$$

---

① 参见 A.H.Nayfeh 和 B.Balachandran, *Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods*, John Wiley & Sons, New York, NY, 1995 年；B.Balechandran 和 E.B.Magrab, 同上，第 38 页。

其特征方程为:

$$\lambda^2 + 2\zeta\lambda + 1 = 0 \tag{9.9}$$

当 $\zeta=0.1$ 时,值可由:

```
lambda = roots([1,0.2,1])
```

求得结果为:$\lambda_1 = -0.1000 + 0.9950i$ 和 $\lambda_2 = -0.1000 - 0.9950i$, $i=\sqrt{-1}$。两者的实部均为负值,表明系统是稳定的,这是欠阻尼系统的特征①。在欠阻尼系统中有:

$$|\mathrm{Re}(\lambda_j)| = \zeta \quad 和 \quad |\lambda_1| = |\lambda_2| = 1$$

例 9.2 在给定初始速度和位移时单自由度系统的振动

下面求式(9.8)表示的线性弹簧－质量－阻尼器系统的自由振动响应。给定初始条件

$$x(0) = x_o$$

$$\frac{\mathrm{d}x(0)}{\mathrm{d}\tau} = v_o$$

以及三种阻尼系数 ζ:(1)$\zeta=0.1$,为欠阻尼情况;(2)$\zeta=1$,为临界阻尼情况;(3)$\zeta=5.0$,为过阻尼情况。系统在初始条件 $x_o=1$ 和 $v_o=1$ 下开始振动。

下面的程序用于求 $0\leqslant\tau\leqslant40$ 范围内的自由振动响应:

```
zeta = [0.1,1.0,5.0]; AlphaHat = [0.0,0.0,0.0];
tspan = linspace(0,40,400);
lintyp = char('-k','--k','-.k');
for i = 1:3
 [t,x] = ode45(@FreeOscillation',tspan,[1 1]',[],zeta(i),AlphaHat(i));
 figure(1);
 plot(t,x(:,1),lintyp(i,:));
 hold on
 figure(2);
 plot(x(:,1),x(:,2)lintyp(i,:));
 hold on
end
figure(1)
xlabel('\tau');
ylabel('x(\tau)');
axis([0,40,-1.5,1.5]);
plot([0,40],[0,0],'k-')
legend('\zeta = 0.1','\zeta = 1.0','\zeta = 5.0')
figure(2)
xlabel('Displacement');
ylabel('Velocity');
legend('\zeta = 0.1','\zeta = 1.0','\zeta = 5.0','Location','NorthWest')
axis([-1.5,1.5,-1.5,1.5]);
```

程序的执行结果如图 9.4 所示。上图为位移响应,下图为距离对速度的曲线图。点$(x,\dot{x})$由位移和速度坐标值组成,称为相点。不同初始条件下得到的相轨迹可组成一幅相平面图。

① 参见 D. J. Inman, 出处同上,第 19 页;S.S. Rao, 出处同上,第 130 页;B. H. Tongue, 出处同上,第 25 页;B. Balachandran和E. B. Magrab,出处同上,第 155 页。

式(9.8)确定的线性系统的平衡点对应于相平面图中 $x_1=0$ 和 $x_2=0$ 的点[或点(0.0,0.0)]。由位移响应图上看出，只有欠阻尼情况有振动，对应到相平面图上是螺旋形曲线轨迹。随着时间的增加趋向于点(0.0,0.0)，该点为平衡点，也是一吸引点。在临界阻尼和过阻尼情况下，在系统到达平衡位置的过程中并没有发生振动。对临界阻尼系统，到达平衡位置的时间最短。

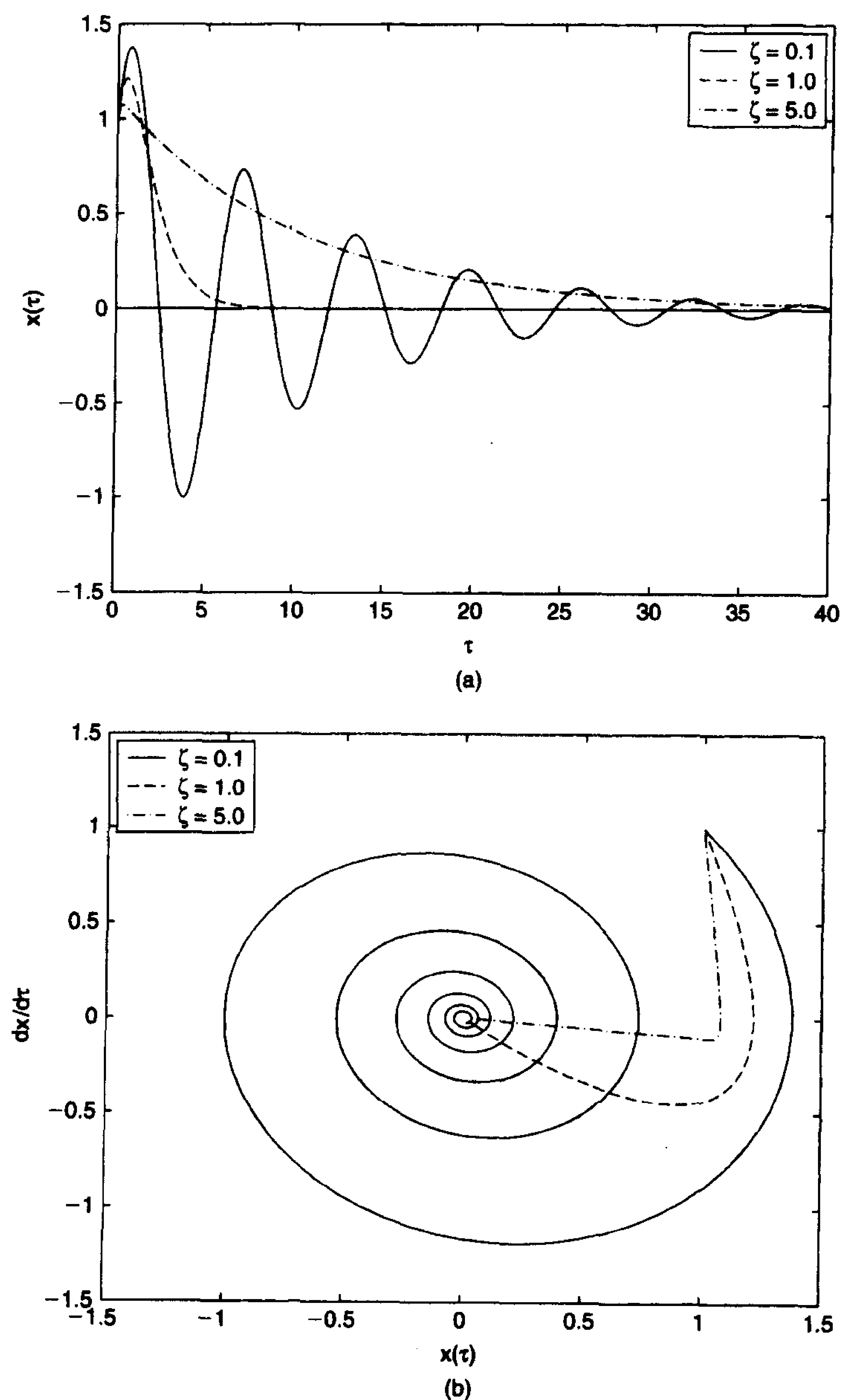


图 9.4　阻尼线性系统自由振动：(a) 位移响应图；(b)相平面图

对数衰减率

现由欠阻尼系统 ζ 的位移响应曲线估计阻尼因子。首先,确定对数衰减率 δ,δ 由下式求得①:

$$\delta = \frac{1}{n}\ln\left(\frac{x_j}{x_{j+n}}\right) \tag{9.10}$$

阻尼因子与 δ 有关,具体定义如下:

$$\zeta = \frac{\delta^2}{\sqrt{4\pi^2 + \delta^2}} \tag{9.11}$$

x_j 和 x_{j+n} 分别为时间 t_j 和 $t_j + nT$ 对应的位移,T 为欠阻尼振动的周期。求 δ 的一种方法为先求不同的极大值(极小值)对应的幅度,然后从产生最大值处开始计算其周期 T。由这些结果可确定 x_j。

例 9.3 利用对数衰减率对阻尼系数进行估计

图 9.5 为一欠阻尼系统的自由振动响应。参照此图,利用下面的函数对系统的阻尼系数进行估计:

```
function[pspace,tref] = period(tdata,xdata,n,nskip)
[xmin0,tmin0] = min(xdata((1 + nskip):length(xdata)));
[xmax0,tmax0] = max(xdata((1 + nskip):length(xdata)));
tmin0 = tmin0 + nskip;tmax0 = tmax0 + nskip;
m = fix(length(tdata)/(abs(tmin0 - tmax0) * 2.0));
while n > m
 n = n - 1;
end
pspace = zeros(n,1);
tref = zeros(n + 1,1);
for j = 1:n
 if tmin0 < tmax0
  tref(1) = tmin0;
  if j == 1
   [xmin 1,tmin 1] = min(xdata(tmax0:length(xdata)));
   pspace(j) = (tmax0 - tmin0) + tmin1;
   tref(j + 1) = tref(j) + pspace(j);
  else
   [xmaxj,tmaxj] = max(xdata(tref(j):length(xdata)));
   [xminj,tminj] = min(xdata(tref(j) + tmaxj:length(xdata)));
   pspace(j) = tmaxj + tminj;
   tref(j + 1) = tref(j) + pspace(j);
  end
 else
  tref(1) = tmax0;
  if j == 1
   [xmax1,tmax1] = max(xdata(tmin0:length(xdata)));
   pspace(j) = (tmin0 - tmax0) + tmax1;
   tref(j + 1) = tref(j) + pspace(j);
  else
```

① 参见 D.J,Inman,出处同上,第 44 页;S.S.Rao,出处同上,第 136 页;B.Balachandran 和 E.B.Magrab,出处同上,第 145 页。

```
        [xminj,tminj] = min(xdata(tref(j):length(xdata)));
        [xmaxj,tmaxj] = max(xdata(tref(j) + tminj:length(xdata)));
        pspace(j) = tmaxj + tminj;
        tref(j+1) = tref(j) + pspace(j);
      end
    end
  end
```

period 函数的输入量为时间参数 *tdata*，对应的响应参数 *xdata*，循环次数 n 以及省略的初始数据点 *nskip*。*period* 函数的输出量包括数组 *pspace* 和 *tref*，分别为波峰(或波谷)间的间隔指标和与之对应的时间指标。函数 max 和 min 用于求波峰(极大值)对应的时间和波谷(极小值)对应的时间以及波峰与波谷之间的时间差。第二行和第三行分别求出第一个波谷和第一个波峰的位置。由于没有用 gradient 函数检查极值是否存在，所以在确定极值点尤其是第一个极值点时可能会产生错误。例如，图9.5中显示的数据，如果 min 函数的起始时间设为 $\tau=0$，则图中第一个数据点被误认为是第一个极小值。为避免此类错误，提供了一个输入参数 *nskip* 以定义数据的起始点。

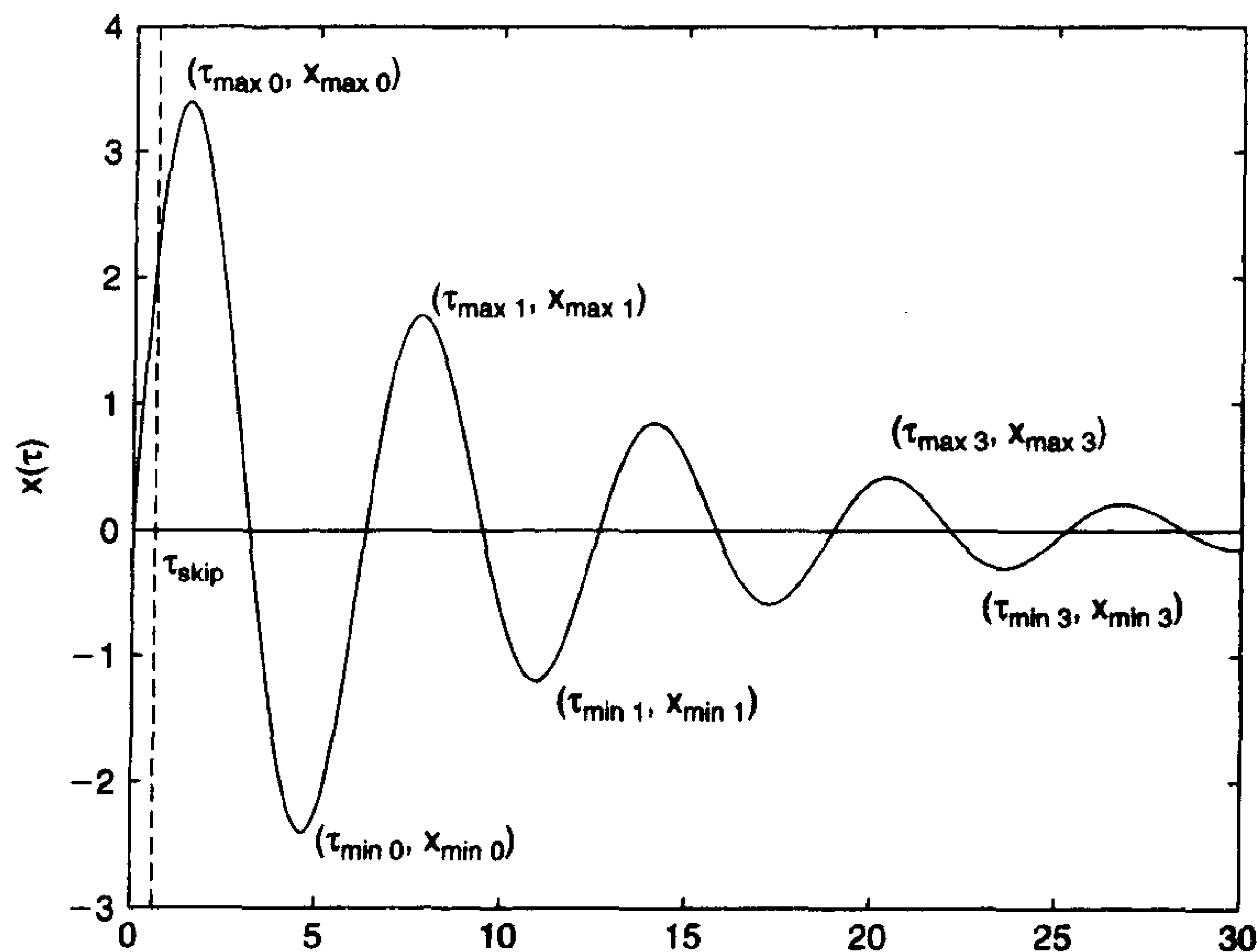


图9.5　欠阻尼系统的自由振动响应

极大值(极小值)与其后继极小值(极大值)之间的间隔对应于阻尼振动的半个周期。在第5行利用这一点估算 *xdata* 中的总循环次数 m。如果定义的输入量 n 大于 m，则 n 设置为小于或等于 m。从第9行开始对数组 *pspace* 和 *tref* 进行初始化，随后在 for 循环内求出极值(极大值或极小值)间的间隔。在循环体内还用到逻辑判断语句以区分第一个极值为极小值的情况和第一个极值为极大值的情况。后一种情况求出波峰之间的间隔指标以及相关的时间指标，对其他情况求出波谷之间的间隔指标以及相关的时间指标。由 **period** 函数求得的极值可以作为确定经过不同循环后阻尼振动的周期的依据。

尽管线性系统的阻尼振动周期在经过不同循环后不会改变，但对于非线性系统却并非如此。接下来的程序中利用 **period** 函数的输出量和式(9.10)估算式(9.8)的系统响应的对数衰减率。系统参数为 $\zeta=0.3$ 和 $\hat{\alpha}=0.0$，响应的起始点为 $x_0=0.0$ 和 $v_0=-10.0$。

```
zeta = 0.3; AlphaHat = 0.0;
tspan = linspace(0.0,40.0,400);
[t,x] = ode45(@FreeOscillation,tspan,[0-10]',[],zeta,AlphaHat);
n = input('Enter number of cycles');
nskip = input('Enter number of initial points to skip');
[pspace,tref] = period(t(:,1),x(:,1),n,nskip);
zeta = zeros(length(pspace),1);
fprintf(1,'Cycle Number Damping Factor \ n')
for j = 1:length(pspace)
 logdec = log(x(tref(j),1)/x(tref(j+1),1));
 zeta(j) = sqrt(logdec^2/(4.0 * pi^2 + logdec^2));
 fprintf(1,'   %3d          %6.4f \ n',j,zeta(j))
end
```

程序前两行用于产生式(9.8)系统的均匀分布的数据，然后调用 **period** 函数求出阻尼因子。在第 10 行和第 11 行分别实现式(9.10)和式(9.11)。给定 $n=4$，$nskip=4$ 时，运行程序，在 MATLAB 命令窗口中显示以下内容：

```
Enter number of cycles 4
Enter number of initial points to skip 4
Cycle Number Damping Factor
     1          0.3008
     2          0.3006
     3          0.2992
     4          0.3004
```

正如所期望的，对一通过黏性阻尼而消耗能量的线性系统来说，衰减是指数衰减，而且在阻尼振动的每次循环中对数衰减率均保持不变。

9.2.3 非线性系统的自由振动

现在研究三个非线性系统的振动。第一个系统中含有一个三次非线性弹簧，即式(9.3)在 $F(\tau)=0$ 且 $\hat{\alpha}\neq 0$ 时的情况。第二个系统中有一非线性阻尼器，第三个系统中有一个无阻尼平面摆。

例 9.4 非线性弹簧系统

对式(9.3)描述的系统运动分三种情况加以检验，三种情况如表 9.2 所示。第一种情况为一线性系统，而第二种和第三种情况为带软弹簧的非线性系统。前两种情况下的初始条件相同，第三种情况则不同。

表 9.2 三种情况下的参数和初始条件

情况	系统类型	$\hat{\alpha}$	ζ	$x(0)=x_o$	$dx(0)/d\tau=v_o$
1	线性	0.00	0.20	-2.00	2.00
2	非线性	-0.25	0.20	-2.00	2.00
3	非线性	-0.25	0.20	-2.00	2.31

9.2.1 节创建的函数 **FreeOscillation** 可以描述式(9.3)系统。下面的程序针对上述三种情况产生了不同的响应输出。此外，在估算阻尼因子的同时还求出响应曲线在不同循环中的周期。

```
zeta = 0.2; AlphaHat = [0.00  -0.25  -0.25];
xo = [ -2.00  -2.00  -2.00];  vo = [2.00 2.00 2.31];
tspan = linspace(0.0,30.0,401);
lintyp = char('-k','--k','-k');
options = odeset('RelTol',1e-8,'AbsTol',[1e-8 1e-8]);
d = char('Linear:x _ o = -2 v _ o = 2 \ alpha = 0',...
   'Nonlinear:x _ o = -2 v _ o = 2 \ alpha = -0.25',...
   'Nonlinear:x _ o = -2 v _ o = 2.31 \ alpha = -0.25');
for i = 1:3
   [t,x] = ode45(@FreeOscillation,tspan,[xo(i) vo(i)]',options,zeta,AlphaHat(i));
   figure(1)
   plot(t,x(i,:),lintyp(i,:))
   hold on
   figure(2)
   plot(x(:,1),x(:,2),lintyp(i,:))
   hold on
   fprintf(1,['Case' num2str(i)' 'd(i,:)])
   [pspace,tref] = period(t,x(:,1),3,10);
   zetaest = zeros(length(pspace),1);
   fprintf(1,' \ n Cycle No.     Period     Damping Factor \ n')
   for j = 1:length(pspace)
    per(j) = t(tref(j+1)) - t(tref(j));
    logdec = log(x(tref(j),1)/x(tref(j+1),1));
    zetaest(j) = sqrt(logdec^2/(4.0 * pi^2 + logdec^2));
    fprintf(1,'   %3d       %8.3f       %6.4f \ n',j,per(j),zetaest(j))
  end
 end
 figure(1)
 xlabel(' \ tau')
 ylabel('x( \ tau)')
 axis([0.0,30.0,-3.0,3.0])
 legend(d(1,:),d(2,:),d(3,:))
 figure(2);
 xlabel('x( \ tau)')
 ylabel('dx/d \ tau')
 legend(d(1,:),d(2,:),d(3,:))
 axis([-2.0,3.0,-2.0,3.0])
```

第5行定义了积分过程中用到的误差值以确保数值误差不至于产生无效解。

严格来说,对数衰减率在估算非线性系统的阻尼因子时不是很有效。这里可以说明基于非线性系统的“大”振幅估算的阻尼因子会产生错误解,而基于“小”振幅估算的阻尼因子可得出合理解。并且,此结果也说明非线性系统的振动周期与响应振幅有关。该程序运行结果在 MATLAB 命令窗口上输出每种情况下经过不同循环时阻尼因子的估算值。

```
Case  1  Linear:x _ = -2 v _ o = 2 alpha = 0
Cycle No.       Period       Damping Factor
    1           6.450          0.2002
    2           6.450          0.2004
    3           6.375          0.1996
Case  2  Nonlinear:x _ o = -2 v _ o = 2 alpha = -0.25
Cycle No.       Period       Damping Factor
```

```
      1          7.200          0.2319
      2          6.450          0.2015
      3          6.450          0.2005
  Case  3  Nonlinear: x _ o = -2 v _ o = 2.31 alpha = -0.25
  Cycle No.      Period        Damping Factor
      1          9.000          0.2589
      2          6.450          0.2017
```

三种情况下的自由振动图如图 9.6 所示。

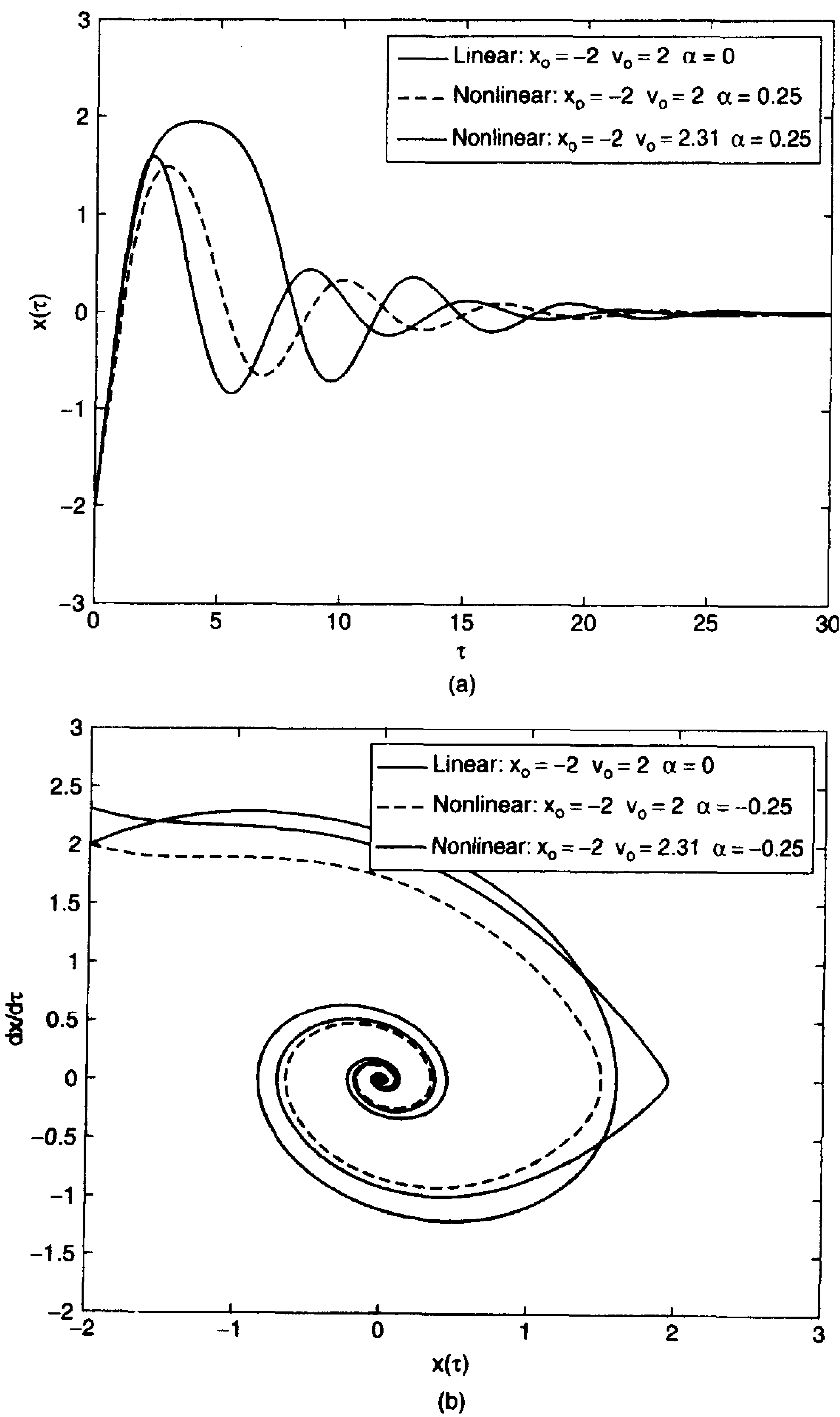


图 9.6 阻尼、线性和非线性振动固有响应:(a) 位移图;(b) 相平面图

第一种情况为线性系统，其阻尼振动周期在每次循环时实质上保持不变。第二种和第三种情况均为非线性系统，其阻尼振动周期和阻尼因子在第 1 个循环与后继循环内的估算值有显著差异。正如图中第三种情况中第 1 个周期所示，当振幅“大”时，非线性效果非常明显。

在第二种和第三种情况中，随着振幅变“小”而逐渐接近于线性系统（第一种情况）。

在相平面图中，与线性情况一样，非线性系统从初始条件开始的运动轨迹趋近于稳定的平衡位置(0.0,0.0)处。相平面图中振动曲线为螺旋形表明对应的非线性系统的振动为欠阻尼振动。在稳定的平衡位置附近的“小”振幅处，非线性系统应表现为线性系统。对初始条件不同的两种响应进行分析，可以看出在位移图上，第三种情况的第 1 个极值附近的曲线并不是典型的线性系统响应曲线。此时的曲线轨迹接近于系统的不稳定平衡位置(2.00,0.00)处，系统的振动受到影响。与线性系统不同，非线性系统可以有多个平衡位置，但并不是所有的平衡位置都稳定①。

非线性阻尼系统

如图 9.7 所示的弹簧－质量系统，考虑系统有干摩擦，其方程为：

$$\frac{d^2x}{d\tau^2}+x+d\ \mathrm{signum}(dx/d\tau)=0 \tag{9.12}$$

式中，常量 $d=\mu mg/k$，μ 为摩擦系数，mg 为物体的重量，k 为线性弹簧的弹性系数。干摩擦力是速度的分段常数函数，在式(9.12)中用 signum 表示。速度为正值时 signum 为 +1，速度为负值时 signum 为 -1。由于干摩擦力随速度的不同而呈现非线性变化，所以系统为一非线性系统。在系统的振动过程中，如果弹簧的弹性力不能克服干摩擦力，系统将停止振动。即当

$$\frac{dx}{d\tau}=0 \qquad 且 |x|\leqslant d \tag{9.13}$$

时，系统停止振动。式(9.12)表示的非线性系统具有多个平衡位置，这些平衡位置在相空间中形成连结点($-d$,0)和点(d,0)的一条直线。

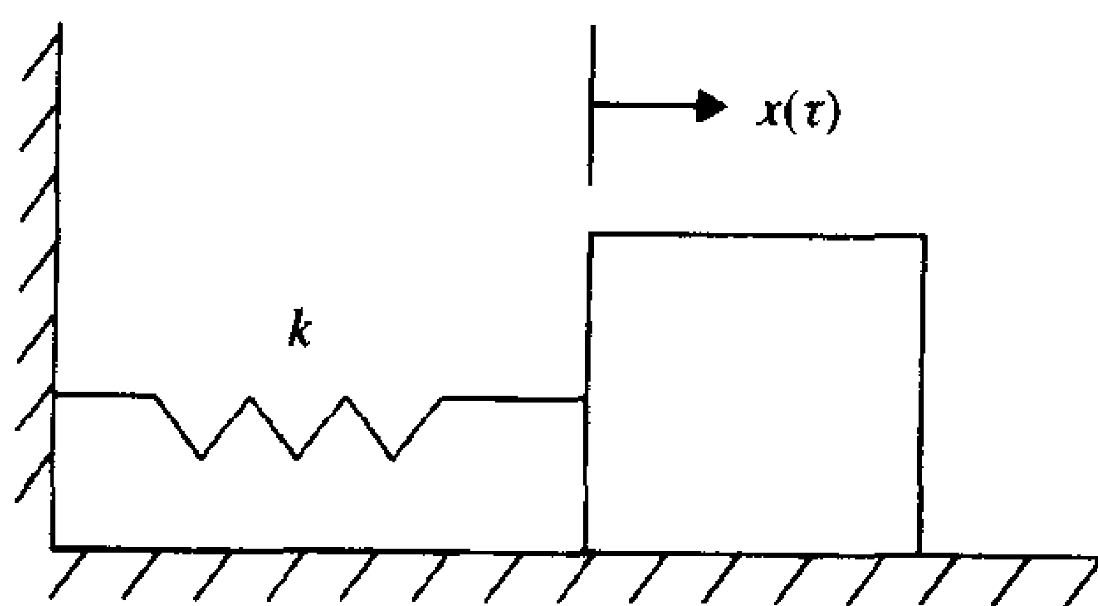


图 9.7　有干摩擦的弹簧－质量系统

例 9.5　库仑阻尼系统的响应

根据系统在 $dx/d\tau>0$ 范围内及 $dx/d\tau<0$ 范围内均为线性这一点，可以求出式(9.12)的封闭解②。但这里将用 ode45 求其数值解。首先引入变量：

① 参见 A. H. Nayfeh 和 B. Balachandran，出处同上。

② 参见 D. J. Inman，出处同上，2.7 节；S. S. Rao，出处同上，2.7 节。

$$x_1 = x$$

$$x_2 = \frac{\mathrm{d}x}{\mathrm{d}\tau}$$

式(9.12)变为一阶方程形式:

$$\frac{\mathrm{d}x_1}{\mathrm{d}\tau} = x_2$$
$$\frac{\mathrm{d}x_2}{\mathrm{d}\tau} = -x_1 - d\ \mathrm{signum}(x_2) \tag{9.14}$$

下面创建子函数**FreeOscillation2**以表示式(9.13)和式(9.14)。在 $d=0.86$,初始条件为(a)(3.0,0.0)及(b)(5.0,0.0)时,求得式(9.12)的数值解。程序如下:

```
function Coulomb
d=0.86;xo=[3.0,5.0];vo=[0.0,0.0];
tspan=linspace(0,12,120);
options=odeset('AbsTol',[le-3,le-3]);
lintyp=char('--k','-k');
for i=1:2
  [t,x]=ode45(@FreeOscillation2,tspan,[xo(i) vo(i)]',options,d);
  figure(1);
  plot(t,x(:,1),lintyp(i,:));
  hold on
  figure(2);
  plot(x(:,1),x(:,2),lintyp(i,:));
  hold on
end
figure(1)
xlabel('\tau');
ylabel('x(\tau)');
axis([0.0,12.0,-4.0,6.0]);
plot([0 12],[0 0],'k-');
legend(['x_o='num2str(xo(1))],['x_o='num2str(xo(2))])
plot([0,12],[0,0],'k-')
figure(2)
xlabel('x(\tau)');
ylabel('dx/d\tau');
text(2.5,0.5,'(3.0,0.0)');
text(4.5,0.5,'(5.0,0.0)');
plot([-4 6],[0 0],'k-',[0 0],[-6 4],'k-')
axis([-4.0,6.0,-6.0,4.0]);

function xdot=FreeOscillation2(t,x,d)
if abs(x(1))<=d & x(2)==0.0
  xdot=[0;0];
else
  xdot=[x(2);-d*sign(x(2))-x(1)];
edn
```

以上程序定义每种状态的绝对误差大于默认值1.0e-6,从而加快计算速度。执行结果如图9.8所示。该系统在两个不同位置处停止振动,到达不同停止位置的时间也不同。这个例子说明非线性系统的长时间响应与初始条件有关。与之相反,一线性阻尼系统的渐近响应与初始条件无关。

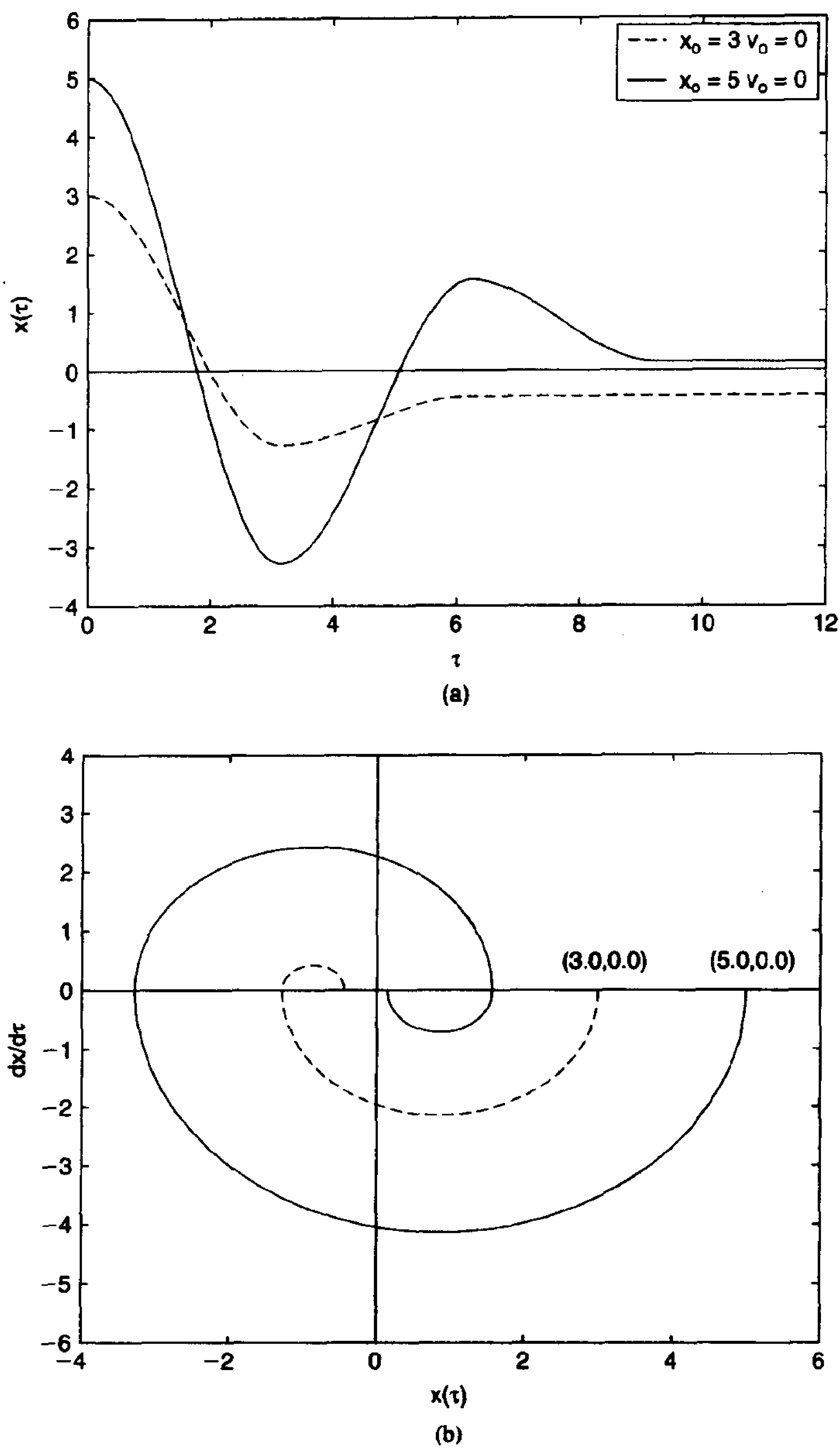


图9.8 有干摩擦振动的固有响应:(a) 位移图;(b) 相平面图

平面摆

图9.9所示的无阻尼平面摆运动方程式如下[①]:

$$\frac{d^2\theta}{d\tau^2}+\sin\theta=0 \tag{9.15}$$

θ 为摆运动的角坐标,$\tau = t\sqrt{g/L}$ 是无量纲时间,g 是万有引力加速度,L 为摆的长度。对

① 参见 D.T.Greenwood,出处同上;B.Balachandran 和 E.B.Magrab,出处同上。

式(9.15)的第一次积分得:

$$F(\theta_1,\theta_2)=\frac{1}{2}\theta_2^2-\cos\theta_1 \tag{9.16}$$

其中,

$$\theta_1=\theta \quad 和 \quad \theta_2=\frac{d\theta}{d\tau}$$

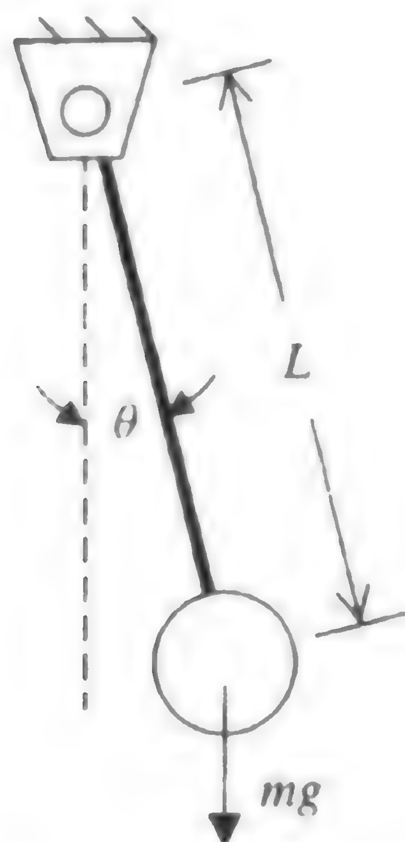


图 9.9 平面摆

用以下程序产生式(9.16)的函数图形如图 9.10 所示。图中谷底最低点与系统稳定平衡位置有关,$\theta=\pm\pi$ 处的峰值与系统不稳定平衡位置有关。

```
thetal1 = linspace(-2.0 * pi,2.0 * pi,50];
theta2 = linspace(-2.0 * pi,2.0 * pi,50];
[T1,T2] = meshgrid(theta1,theta2);
F = T2.^2/2-cos(T1);
meshc(T1,T2,F);
axis([ - 2.0 * pi,2.0 * pi,-2.0 * pi,2.0 * pi,-5,20]);
xlabe(' \ theta _ 1');
ylabel(' \ theta _ 2');
zlabel('F( \ theta _ 1, \ theta _ 2)');
```

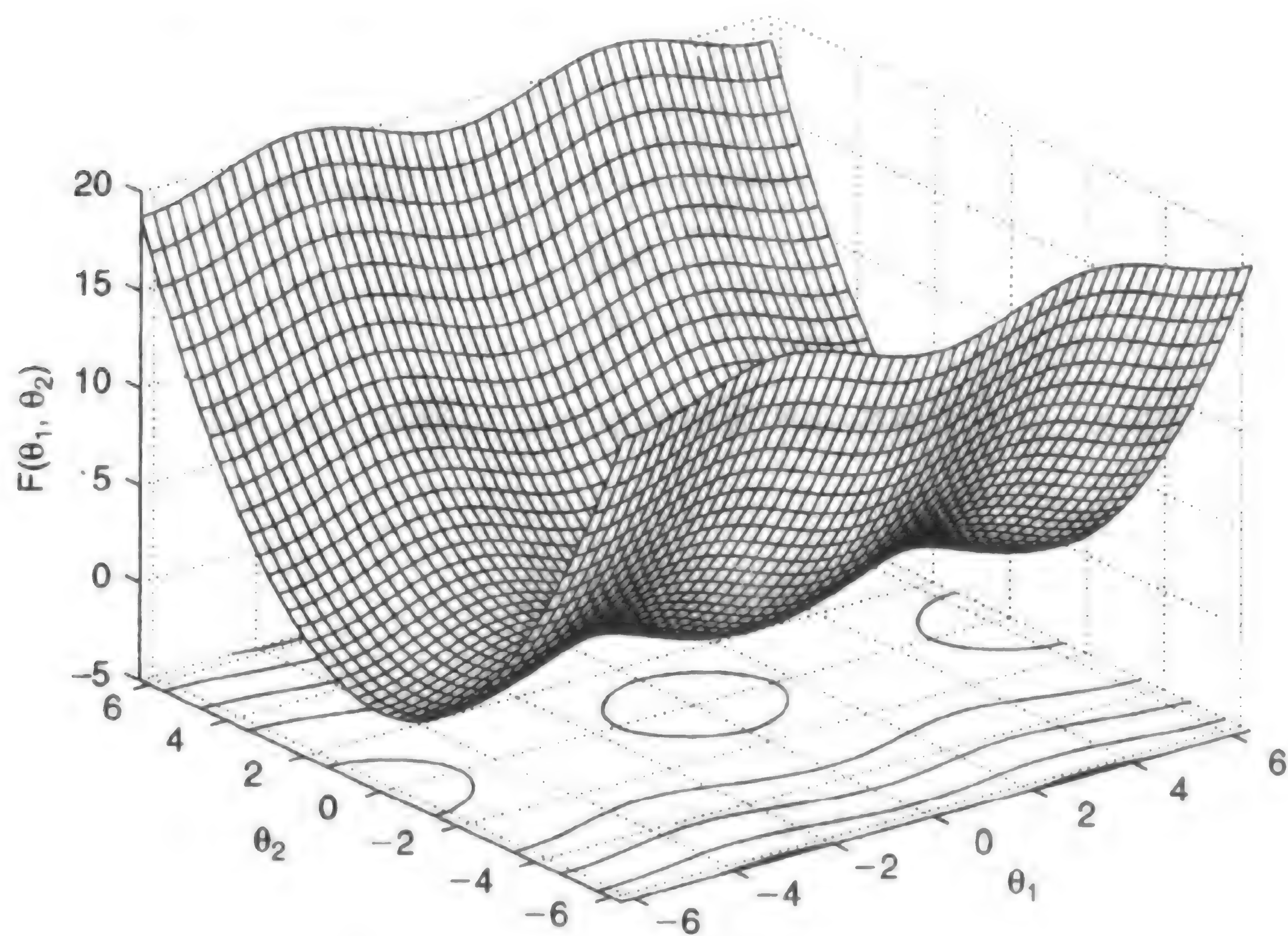


图 9.10 平面摆的表面 $F(\theta_1,\theta_2)$

9.2.4 线性系统与非线性系统的强迫振动

线性系统与非线性系统在谐波激励下的时域响应由 ode45 函数实现,对应的频域响应将用 fft 函数实现。参见 5.4.6 节。

对于式(9.3)确定的系统,令所施力为简谐振动函数,即:

$$F(\tau) = X_o \cos(\Omega \tau) \tag{9.17}$$

其中,Ω 为无量纲激励频率,X_o 为所施力的幅值。选定激励频率和激励振幅,可以求出式(9.3)系统的稳态响应,即有:

$$\lim_{t \to \infty} x(t)$$

并检验在下列不同情况下响应的频谱:(1)$\hat{\alpha} = 0$(线性系统),(2)$\hat{\alpha} \neq 0$(非线性系统)。

在式(9.17)所表示的外力作用下,给定下列参数:(1)初始条件(0,0),(2)$\zeta = 0.4$,(3)$\hat{\alpha} = 1.5$(非线性情况),(4)$\Omega = 3.0$,(5)$X_o = 5.0$。用 ode45 对式(9.3)进行数值积分。激励频率 Ω 选择为系统固有频率的三倍。保存上述数据以用于接下来的例子。程序为:

```
function NonlinearResponse
zeta = 0.4; AlphaHat = [0 1.5];
Omega = 3.0; Xo = 50.0;
tspan = linspace(0,30,6000);
sampint = tspan(2);
options = odeset('RelTol',1e-8,'AbsTol',[1e-8 1e-8]);
for m = 1:2
  [t,x] = ode45(@ForcedOscillator1,tspan,[0 0]',options,zeta,AlphaHat(m),Omega,Xo);
  if m == 1
    figure(1);
    plot(t,x(:,1));
    axis([0 30,-8,8]);
    xlabel('\tau');
    ylabel('x(\tau)');
    yy = x(:,1);
    save 'c:\path\ForcedOscLin.txt' yy -ascii;
else
  figure(2);
  plot(t,x(:,1));
  axis([0 30,-8,8]);
  xlabel('\tau');
  ylabel('x(\tau)');
  yy = x(:,1);
  save 'c:\path\ForcedOseNonLin.txt' yy -ascii;
end
end
function xdot = ForcedOscillator1(t,x,zeta,AlphaHat,Omega,Xo)
xdot = [x(2);-2*zeta*x(2)-x(1)-AlphaHat*x(1)^3+Xo*cos(Omega*t)];
```

其中 *path* 由用户定义。

程序执行结果如图 9.11 所示。由图中看出,线性系统与非线性系统在 $\tau \geq 8$ 时均到达稳态响应。该时间值与指标 $N_{start} = 1600$ 有关。尽管两个稳态响应的周期都等于谐波强制函数的周期,但在频域图中可以清楚地区分两个响应的不同之处。

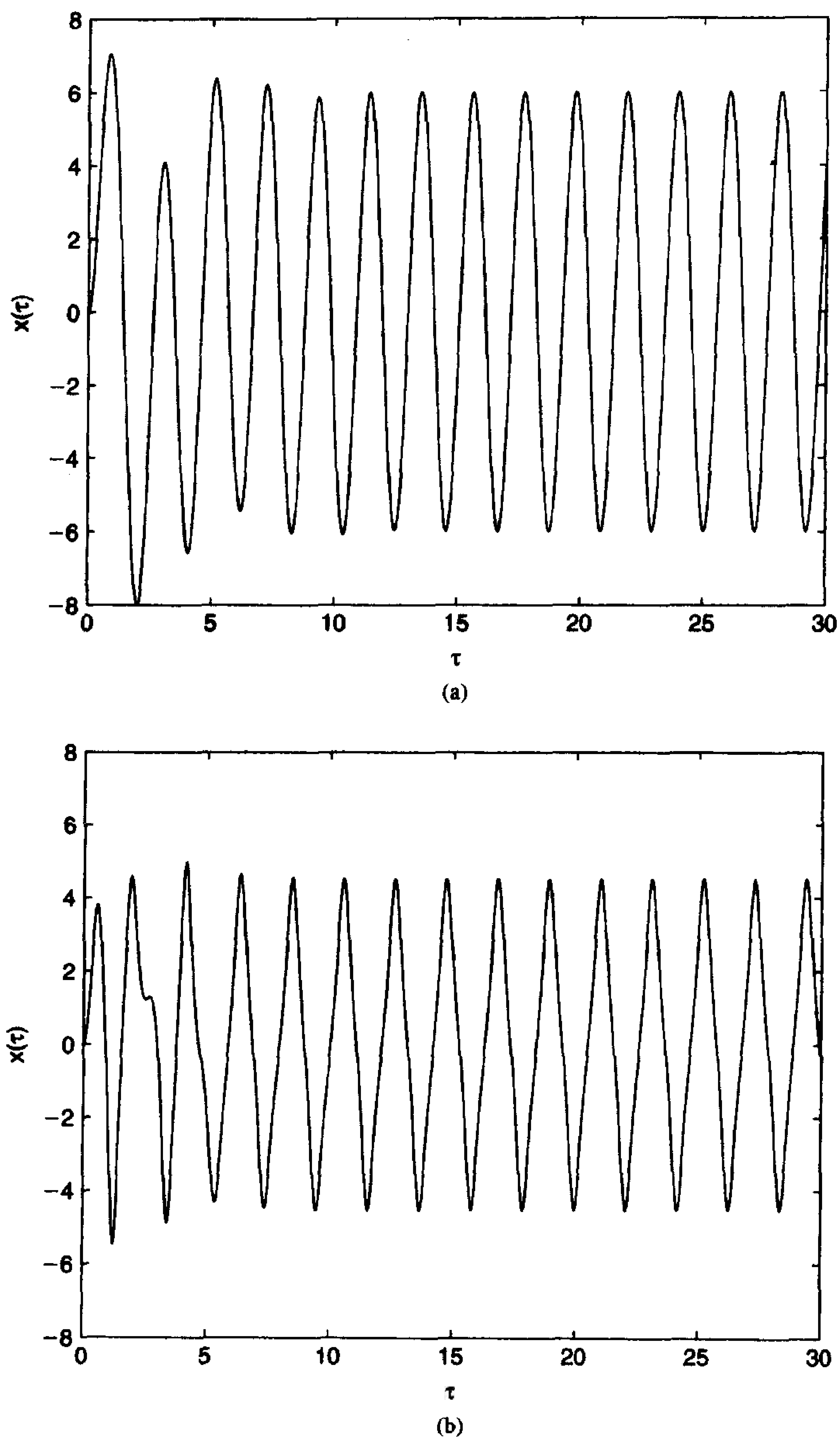


图 9.11　响应图:(a) 线性系统;(b) 非线性系统

为得到频域图中的数据,对 *ForcedOscLin.txt* 保存的线性系统时域图和 *ForcedOscNonlin.txt* 保存的非线性系统时域图的部分数据用 `fft` 函数求解。然后,利用 5.4.6 节的结果可计算出 $\tau \geqslant 8$ 时每个信号的幅频谱。对 *ForcedOscLin.txt* 与 *ForcedOscNonLin.txt* 中数据的采样速率为 $\tau_s = 30/6000 = 0.005$,因此,采样频率(无量纲)为 $f_s = 1/\tau_s = 200$,对基于激励频率 $\Omega = 3$ 的响应,这个值远远超出了必须达到的采样频率,结果不得不截短频谱图,这样仅能显示前 40 个值,而且需要设置 $N_{start} = 3200$, $N = 2^{11} = 2048$。程序为:

```
load 'c:\path\ForcedOscLin.txt';
load 'c:\path\ForcedOscNonLin.txt';
N=2048;Nstart=3200;Fs=200;
f=(Fs*(0:N-1)/N)*2.0*pi;
figure(1)
AmpLin=abs(fft(ForcedOscLin(Nstart:Nstart+N),N))/N;
semilogy(f(1:40),2*AmpLin(1:40));
xlabel('Frequency');
ylabel('Amplitude');
text(3.1,10^4.5,'\Omega');
figure(2);
AmpNonLin=abs(fft(ForcedOscNonLin(Nstart:Nstart+N),N))/N;
semilogy(f(1:40),2*AmpNonLin(1:40));
v=axis;
xlabel('Frequency');
ylabel('Amplitude');
text(3.1,0.5*v(4),'\Omega');
text(9.1,0.2*v(4),'3\Omega');
```

其中,*path* 由用户定义。

执行结果如图 9.12 所示。从图中可以看出,非线性系统的位移响应幅频谱在强制频率 Ω 及其整数倍处获得峰值,其余峰值由弹簧的三次非线性所致。对线性系统而言,只有一个频谱峰值,且与激励频率有关。上例说明非线性系统的响应可以有不同于激励频率的频谱成分。

9.2.5　频率响应、阶跃响应及脉冲响应

下面说明怎样计算和显示线性系统的频率响应曲线。本节中介绍的原理与 9.2.4 节介绍的原理基本相同,本节中给出了控制工具箱中的函数

```
bode, tf, step, impulse, damp
```

的用法。一线性时不变系统(例如,常微分方程)的传递函数可表示为:

$$G(s)=\frac{N(s)}{D(s)}$$

其中,$N(s)$和 $D(s)$为复变量 s 的多项式。已知 $N(s)$和 $D(s)$ 系数数组时,函数

```
sys=tf(N,D)
```

用于定义系统的传递系数。

频率响应函数:

$$G(i\omega)$$

可由

```
bode(tf(N,D),w)
```

计算并画出图形,图形包括 $G(i\omega)$的幅频特性及相频特性图,或者用:

```
[magnitude,phase]=bode(tf(N,D),w)
```

求出幅值及相位的数值解数组。数组 w 或为定义考虑范围内频率的最小值与最大值的两元素单元,或为一角频率数组。

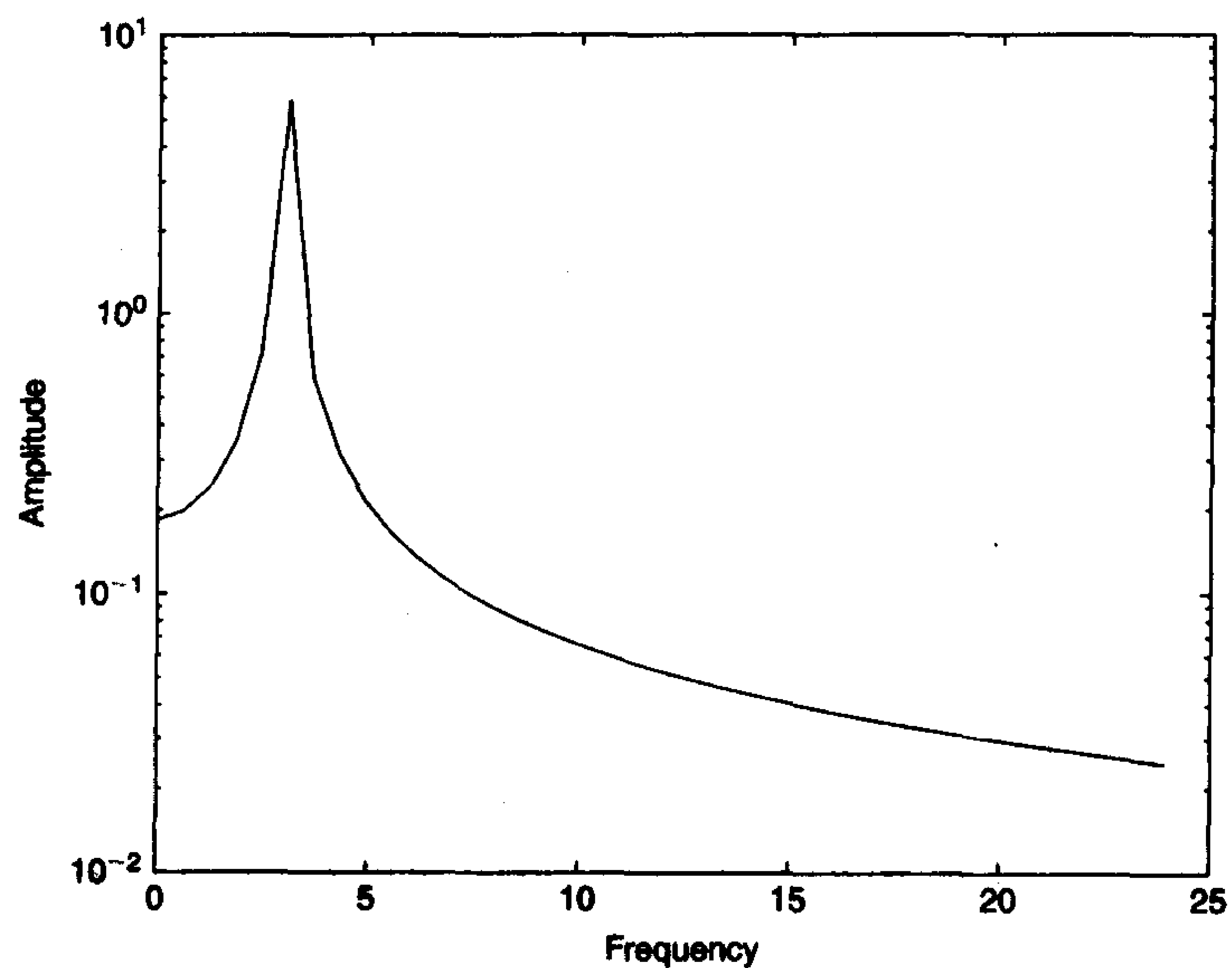
函数 impulse 和 step 分别用于求线性时不变系统由静止到振动的脉冲响应与阶跃响应。

```
impulse(tf(N,D))
```

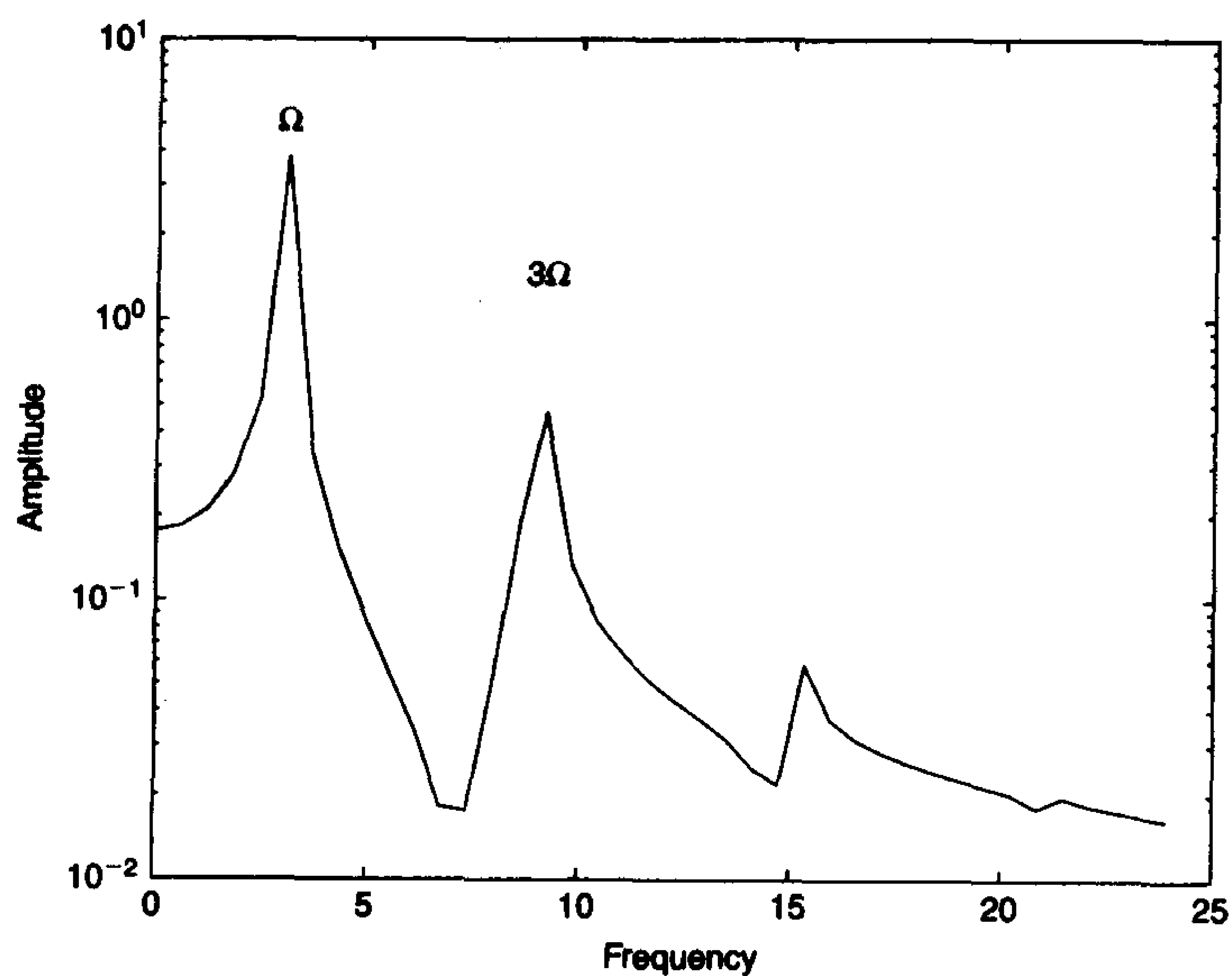
画出 tf 所描述系统的脉冲响应。

```
step(tf(N,D))
```

画出 $t=0$ 时系统的单位阶跃响应。关于这两个函数的其余用法可参考第 10 章。



(a)



(b)

图 9.12 强迫响应的傅里叶分析:(a) 线性系统;(b)非线性系统

函数:

```
[wn,zeta] = damp(tf(N,D))
```

用于求线性时不变系统传递函数的阻尼因子 ζ 和自然频率 ω_n。

例 9.6　频率响应——波特图

为说明这些函数的用法，再考虑式(9.3)在 $\hat{\alpha}=0$ 时表示的系统。对该系统的传递函数做拉普拉斯变换[①]，则得到无量纲传递函数：

$$\frac{\bar{x}(s)}{X_o f(s)}=G(s)=\frac{1}{s^2+2\zeta s+1} \tag{9.18}$$

因此，有：

$$N(s)=1$$
$$D(s)=s^2+2\zeta s+1$$

定义 N 和 D 的数组分别为：

```
N=[0 0 1];
D=[1 2*zeta 1];
```

现用 tf 和 bode 函数将式(9.18)定义的传递函数用于构造频率响应曲线，程序为：

```
zeta=0.2:0.2:1.0;
omega=0.0:0.01:3.0;
for i=1:length(zeta)
  sys=tf([0 0 1],[1 2*zeta(i),1]);
  [mag,phas]=bode(sys,omega);
  figure(1)
  plot(omega,mag(1,:));
  hold on;
  figure(2)
  plot(omega,phas(1,:));
  hold on;
end
figure(1)
xlabel('Frequency ratio');
ylabel('Magnitude');
text(0.8,2.7,'\zeta=0.2');
text(0.8,1.55,'\zeta=0.4');
text(0.8,0.4,'\zeta=1.0');
hold on;
plot([0.0,3.0],[1.0,1.0],'-k');
v=axis;v(2)=2.5;
axis(v)
figure(2)
xlabel('Frequency ratio');
ylabel('Phase(degrees)');
text(0.7,-15.0,'\zeta=0.2');
text(0.5,-80.0,'\zeta=1.0');
hold on;
v=axis;v(2)=2.5;
axis(v)
plot([0.0,3.0],[-90,-90],'-k');
plot([1.0,1.0],[-200,0],'-k');
```

程序执行结果如图 9.13 所示。有些情况下无量纲传递函数的数值指放大倍数。

① 参见 L. Meirovitch，出处同上；参见附录 B；B. Balachandran 和 E. B. Magrab，出处同上，第 116 页。

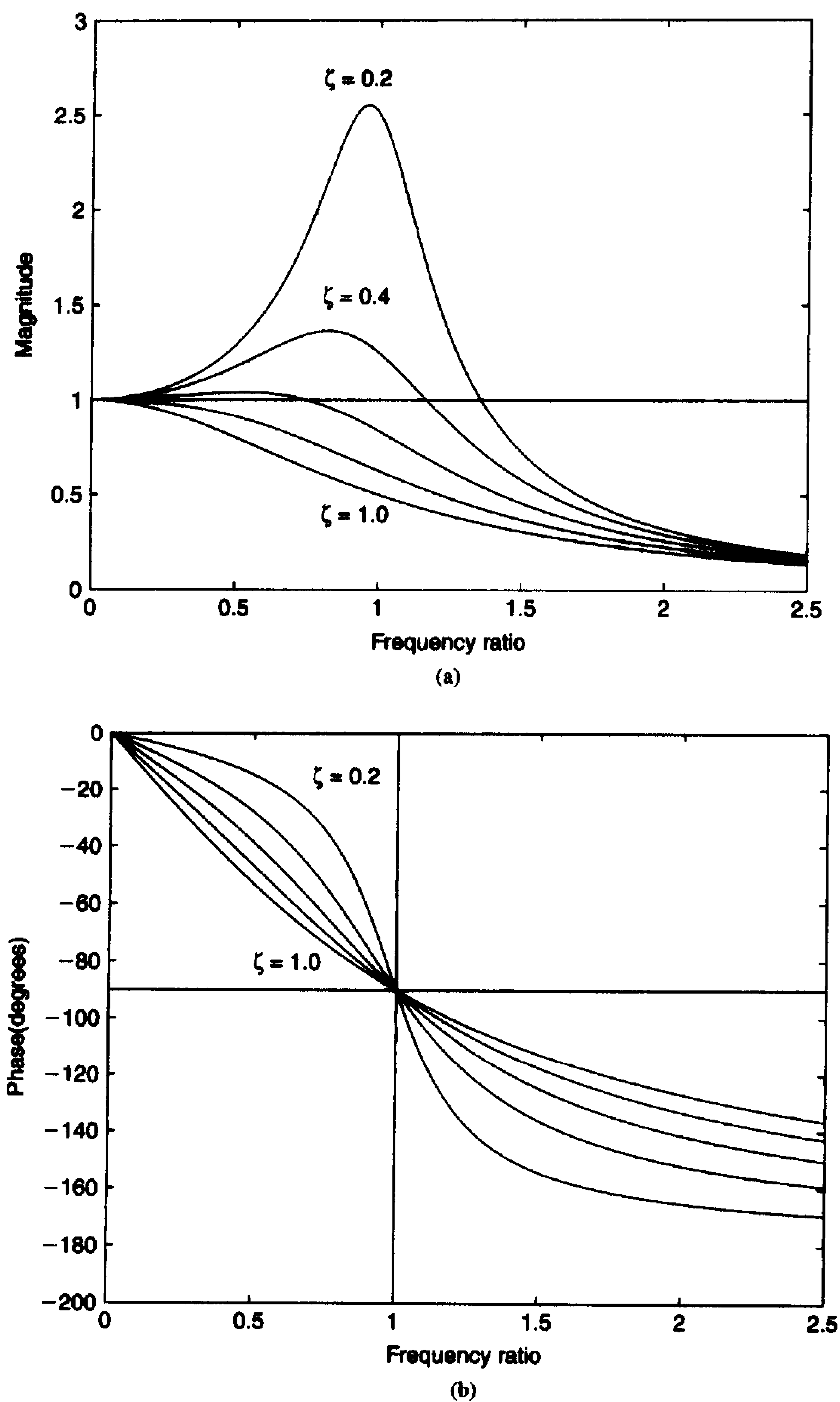


图 9.13 直接激励的弹簧－质量－阻尼器系统的(a)幅频响应;(b)相频特性

例 9.7 脉冲响应

下面求式(9.18)所描述系统的脉冲响应,分别对 $\zeta = 0.1, 1.0$ 和 3.0 所表示的欠阻尼系统、临界阻尼系统及过阻尼系统加以讨论,程序为:

```
zeta = [0.2,1.0,3.0];tfinal = 30;
tdata = linspace(0,tfinal,100);
for i = 1:length(zeta)
  xdata = impulse([0 0 1],[1 2 * zeta(i) 1],tdata);
  plot(tdata,xdata);
  hold on;
```

```
end
axis([0.0,30.0,-0.5,1.0]);
xlabel('\tau');
ylabel('x(\tau)');
text(1.5,0.8,'\zeta=0.2');
text(10.5,0.08,'\zeta=3.0');
```

其执行结果如图 9.14 所示。通过对式(9.18)的时域模型在初始条件(0,1)下求积分,从而求得其自由振动响应,也可获得图 9.14 的结果。

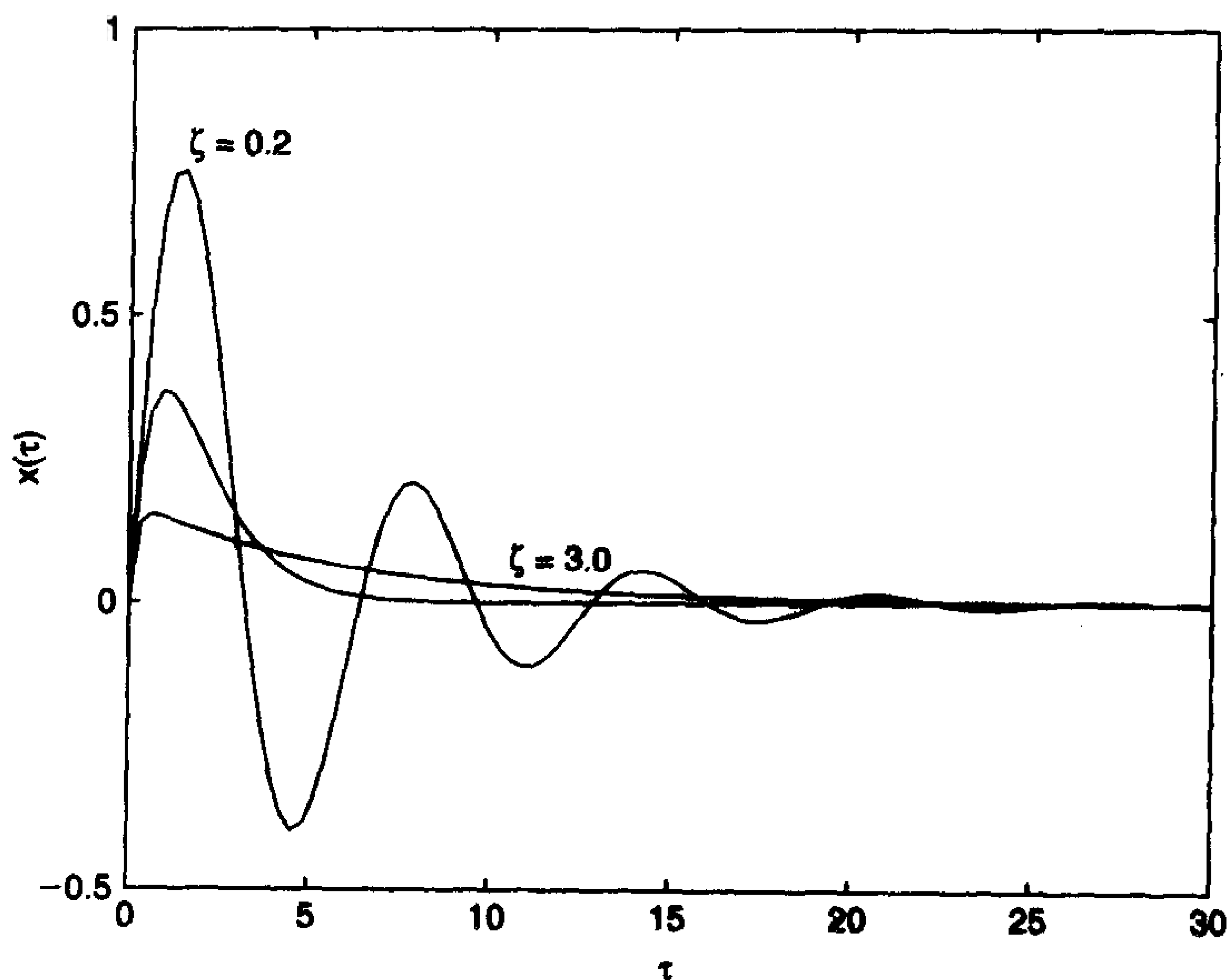


图 9.14 欠阻尼系统、临界阻尼系统、过阻尼系统的脉冲响应

例 9.8 阶跃响应

下面求式(9.18)系统的阶跃响应,分别对 $\zeta=0.2,1.0$ 及 3.0 所表示的欠阻尼系统、临界阻尼系统及过阻尼系统加以讨论。程序为:

```
zeta=[0.2,1.0,3.0];tfinal=30;
tdata=linspace(0,tfinal,100);
for i=1:length(zeta)
  xdata=step([0,0,1]),[1,2*zeta(i),1],tdata);
  plot(tdata,xdata,'k-');
  hold on;
end
xlabel('\tau');
ylabel('x(\tau)');
text(5.0,1.15,'\zeta=0.2');
text(2.7,0.70,'\zeta=1.0');
text(5.0,0.50,'\zeta=3.0');
```

结果如图 9.15 所示。欠阻尼系统的响应在到达稳态之前,经过稳定位置上下摆动,而在临界阻尼与过阻尼情况下却没有摆动。临界阻尼情况下,系统很快就到达稳定位置。

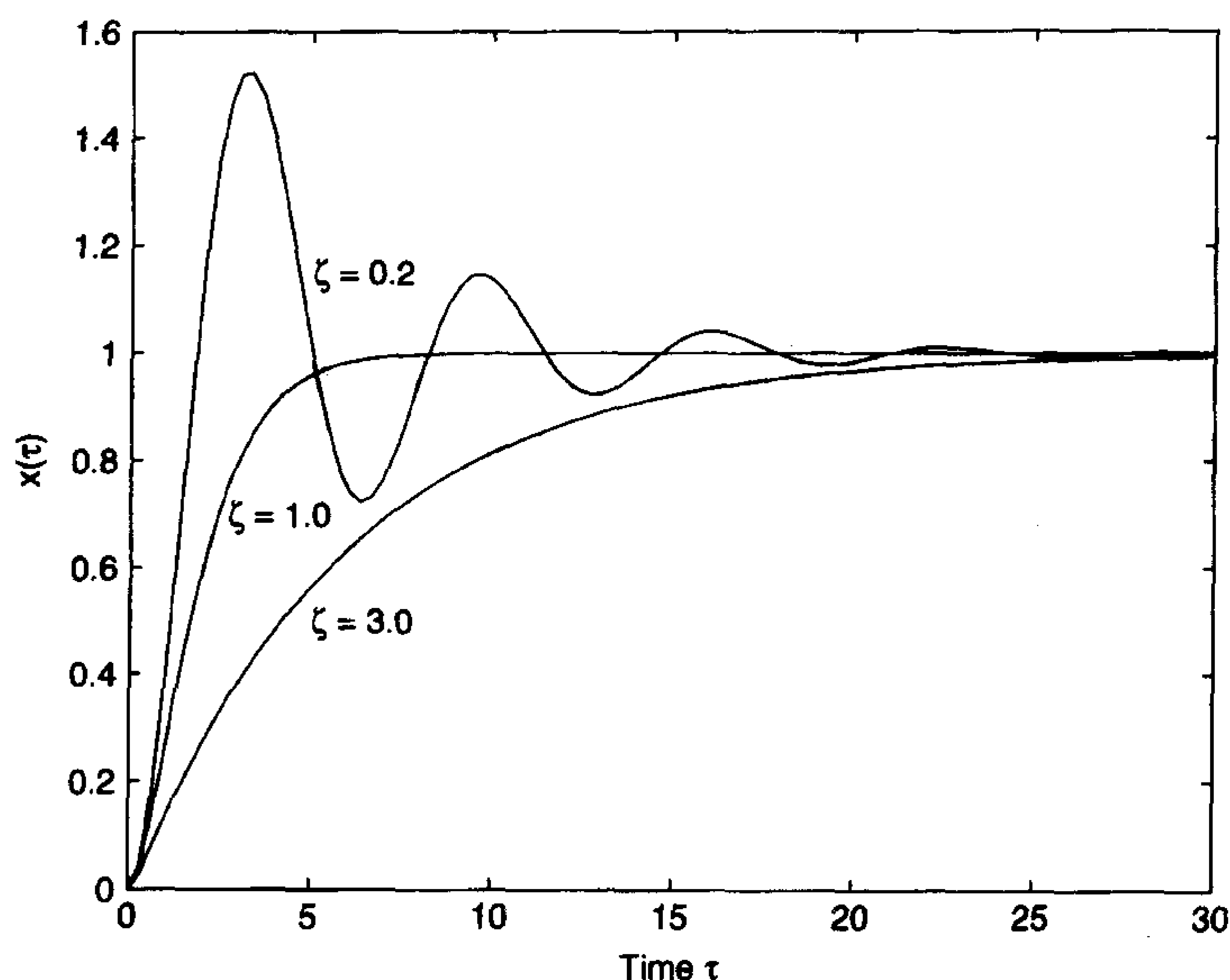


图 9.15 欠阻尼系统、临界阻尼系统及过阻尼系统的阶跃响应

例 9.9 阻尼振动器的 ω_n 和 ζ 的估计

下面将求式(9.18)系统在 $\zeta=0.3$ 时的自然频率与衰减。首先,令系统模型中的 $\zeta=0.3$,然后再用 damp 函数求其值。damp 函数由传递函数的极点求出阻尼因子以及与之相关的自然频率。实现语句为:

```
sys = tf([0 0 1],[1 2*0.3 1]);
[w,zeta] = damp(sys)
```

执行结果为 $\omega_n=1$ 和 $\zeta=0.3$。

例 9.10 频率响应函数的曲线拟和

由式(9.18)得频率响应函数为:

$$G(i\omega)=\frac{A_o}{(1-\omega^2)+2i\zeta\omega} \tag{9.19}$$

其幅度为:

$$|G(i\omega)|=\frac{A_o}{\sqrt{(1-\omega^2)^2+(2\zeta\omega)^2}} \tag{9.20}$$

图 9.16 是对实验数据用式(9.20)进行拟和的曲线。图中的数据点用空心圆表示,曲线用实线表示。使用了优化工具箱的 MATLAB 函数 lsqcurvefit。参数估计值为 $\zeta=0.15$,$\omega_n=1.5$ rad/s。由曲线拟和实现参数估计的程序如下:

```
hh = inline('x(3)./sqrt((1-(w/x(1)).^2).^2 + (2 * x(2) * w/x(1)).^2)','x','w');
xo = [1.3,0.1,0.01];
ww = [0.0,0.0769,0.1538,0.2308,0.3077,0.3846,0.4615,0.5385,...
  0.6154,0.6923,0.7692,0.8461,0.9231,1.000,1.077,1.154,...
  1.231,1.308,1.385,1.462,1.538,1.615,1.692,1.769,1.846...
  1.923,2.000,2.077,2.154,2.231,2.308,2.384,2.461,2.538,...
  2.615,2.692,2.769,2.846,2.923,3.000];
```

```
hr=[0.0141,0.01383,0.01414,0.01418,0.01468,0.01504,0.01532,…
  0.01550,0.01666,0.01784,0.01879,0.01958,0.02156,0.02354,…
  0.02616,0.02956,0.03425,0.03931,0.04456,0.04707,0.04472,…
  0.03888,0.03232,0.02696,0.02179,0.01844,0.01621,0.01381,…
  0.01191,0.0109,0.01002,0.0089,0.0083,0.0073,0.0070,0.0060,…
  0.0054,0.0052,0.00514,0.0043];
x=lsqcurvefit(hh,xo,ww,hr);
w=linspace(0,3,200);
plot(w,hh(x,w),'k-',ww,hr,'ko')
xlabel('\omega')
ylabel('|G(i\omega)|')
```

程序中的函数为实现曲线拟和的函数，式(9.20)用 **hh** 表示，参量 x_o 包括初始估计值 ω_n，ζ 与 A_o。

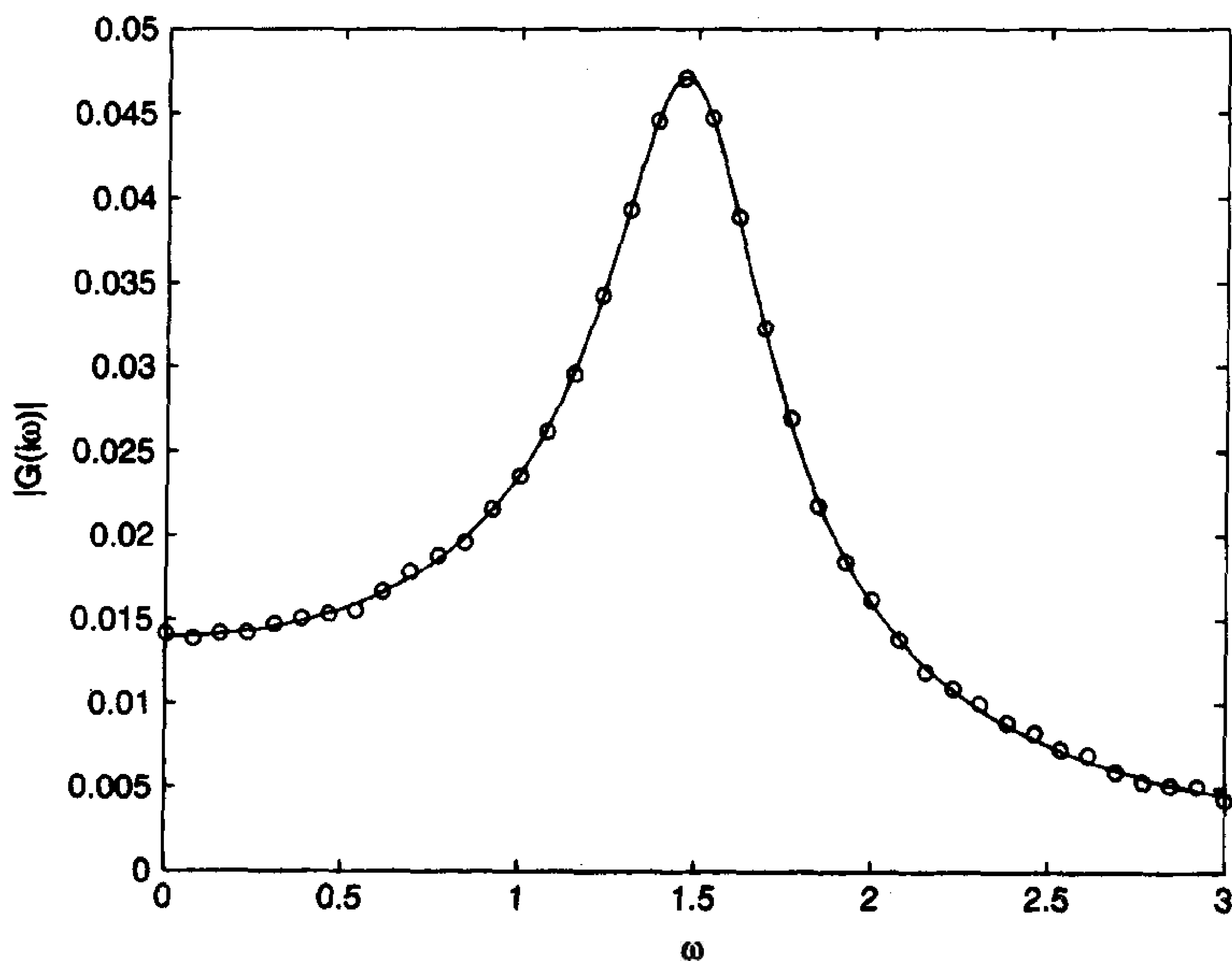


图 9.16 频率响应的曲线拟合

例 9.7 至例 9.10 使用了控制工具箱中的不同函数来研究单自由度系统，也可以将其用于研究多自由度系统，如 9.3 节所述。

9.2.6 机械工具的振动

图 9.17 所示工具的振动方程为①：

$$\frac{\mathrm{d}^2}{\mathrm{d}\tau^2}+\left(\frac{1}{Q}+\frac{K}{k\Omega}\right)\frac{\mathrm{d}x}{\mathrm{d}\tau}+\left(1+\frac{k_1}{k}\right)x-\mu\frac{k_1}{k}x(\tau-1/\Omega)=0 \tag{9.21}$$

其中，无量纲时间 $\tau=\omega_n t$，并且：

$$\Omega=\frac{N}{2\pi\omega_n}\quad \omega_n=\sqrt{\frac{K}{m}}\quad Q=\frac{1}{2\zeta}\quad 2\zeta=\frac{c}{m\omega_n} \tag{9.22}$$

① 参见 B. Balachandran 和 E. B. Magrab，出处同上，4.5 节。

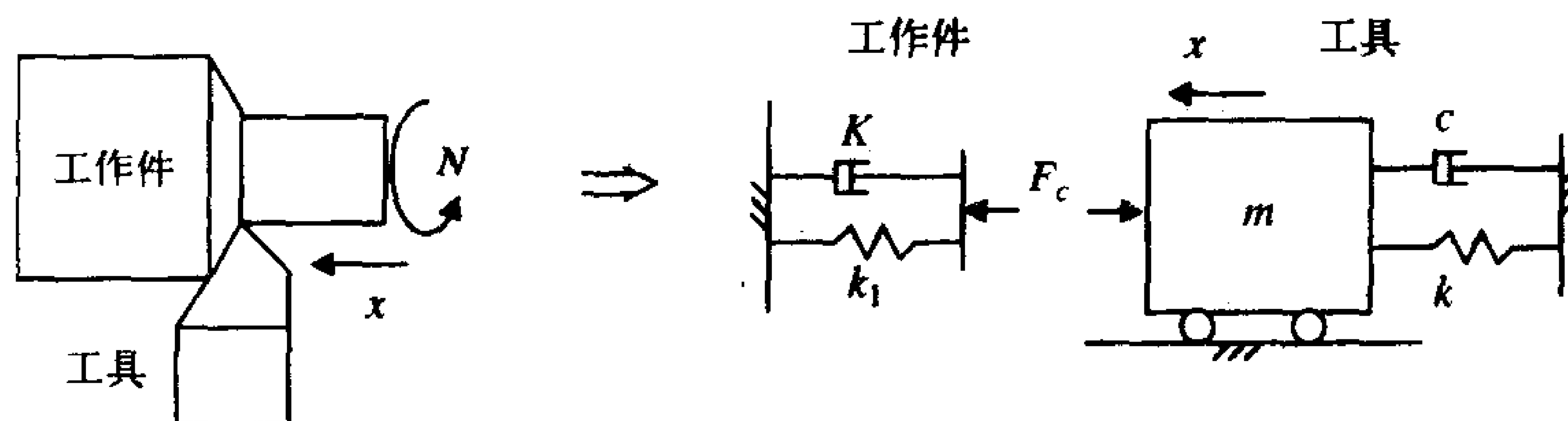


图 9.17　工具和工作件转动时的模型

下面研究振动的可能性。假设有一形式如下：

$$x = Ae^{\lambda\tau}$$

代入式(9.21)得特性方程：

$$\lambda^2 + \left(\frac{1}{Q} + \frac{K}{k\Omega}\right)\lambda + 1 + \frac{k_1}{k}(1 - \mu e^{-\lambda/\Omega}) = 0 \tag{9.23}$$

为找稳定边界条件值，令 $\lambda = i\omega$，将 λ 代入式(9.23)，得：

$$\frac{1}{Q} + \frac{K}{k\Omega} + \frac{\mu k_1}{k}\frac{\sin(\omega/\Omega)}{\omega} = 0$$
$$\omega^2 = 1 + \frac{k_1}{k}(1 - \mu\cos(\omega/\Omega)) \tag{9.24}$$

式(9.24)中，量 K/k，μ，k_1/k 为已知，无量纲轴速度 Ω 在特定范围内变化。对每个 Ω，用函数 fzero 对式(9.24)中的第二个式子运用数值方法求 ω，然后将 Ω 和 ω 代入式(9.24)的第一个式子，求得满足方程的正 Ω 值，即使 Ω 和 ω 满足下式：

$$\frac{1}{Q} = -\frac{K}{k\Omega} - \frac{\mu k_1}{k}\frac{\sin(\omega/\Omega)}{\omega}$$

例 9.11　转动操作的稳定性图

Ω 对 Q 的关系图可以显示出系统稳定或不稳定的区域。在 $K/k = 0.0029$，$k_1/k = 0.0785$ 及 $\mu = 1$ 条件下对式(9.24)运用数值方法求解的程序如下，执行结果如图 9.18 所示。

```
chat = inline('1-w.^2 + k1k * (1-u * cos(w/Ob))','w','u','k1k','Ob');
k1k = 0.0785;Kk = 0.0029;u = 1;
Ob = linspace(0.03,0.5,300);
opt = optimset('Display','off');
L = length(Ob);
for n = 1:L
  w(n) = fzero(chat,[0.8 1.2],opt,u,k1k,Ob(n));
end
xx = -1./(Kk./Ob + u * sin(w./Ob)./w * k1k);
indx = find(xx > = 0);
plot(Ob(indx),xx(indx),'k-')
hold on
a = axis;a(4) = 50;
axis(a)
fill(Ob(indx),xx(indx),'c')
B = sqrt(2) * sqrt(1 + k1k - sqrt(1 + 2 * k1k + (k1k^2) * (1 - u^2)));
Om = 1./(B-Kk./Ob);
ind = find(Ob < 0.05);
plot(Ob(ind(end):L),Qm(ind(end):L),'k--')
xlabel('\Omega')
ylabel('Q')
```

```
text(.22,40,'Unstable')
text(.15,5,'Stable')
text(0.02,10,['K/k = num2str(Kk,5)])
text(0.02,6,['k _ 1/k = 'num2str'(k1k,5)])
grid on
```

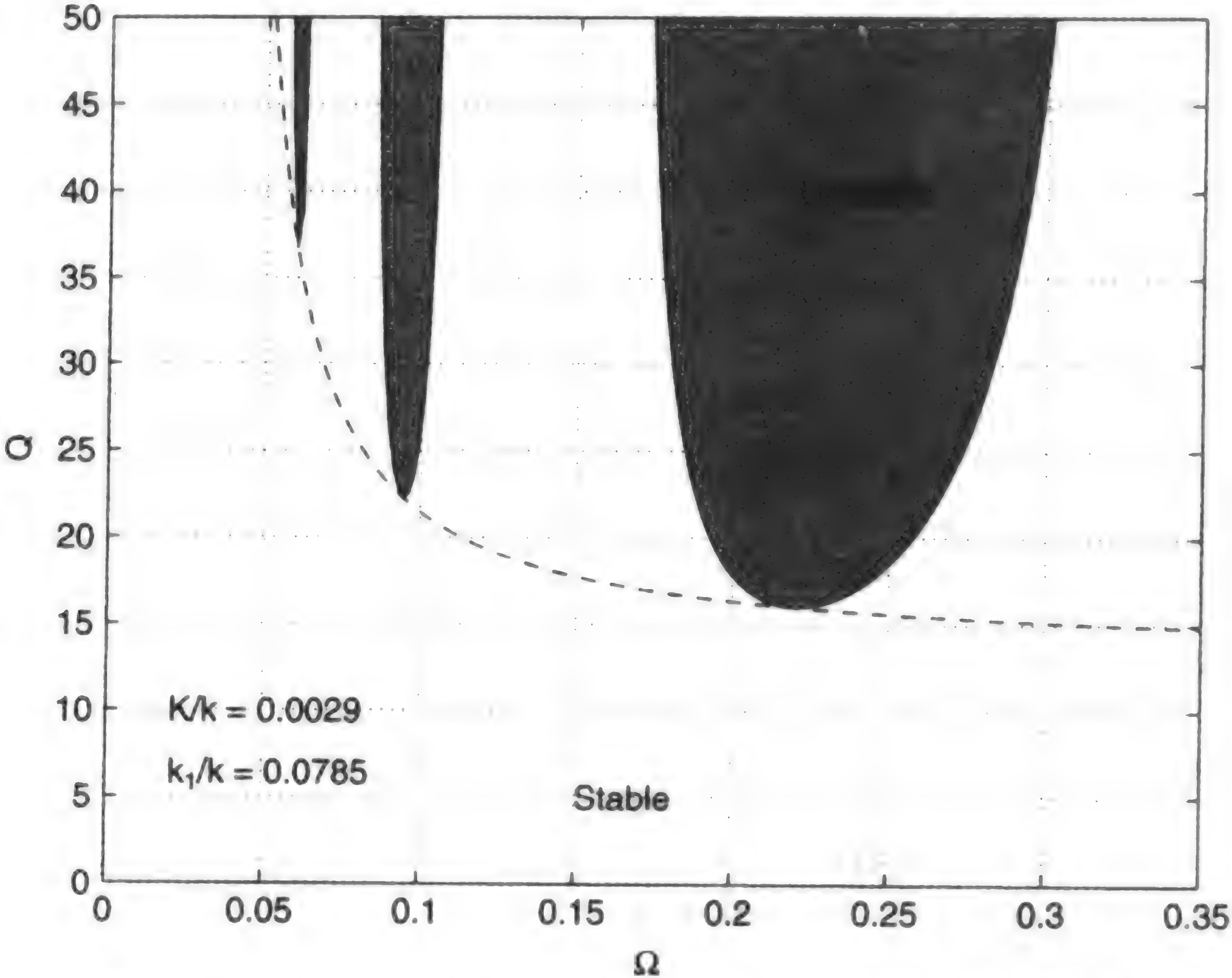


图 9.18　$K/k=0.0029$, $k_1/k=0.0785$ 及 $\mu=1$ 时的稳定性图

9.3　多自由度系统

9.3.1　自由振动

下面分析三类不同的问题，对每类问题都需要求出系统的特征值。

1. 主惯性矩问题
2. 转动刚体的稳定性问题
3. 多自由度系统的固有频率问题

主惯性矩问题

如图 9.19 所示，一刚体有三个转动自由度。其相关的转动惯量矩阵有以下形式：

$$I_{rot}=\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{9.25}$$

其中，各个惯性矩的定义参照图 9.19 所示的坐标系统。

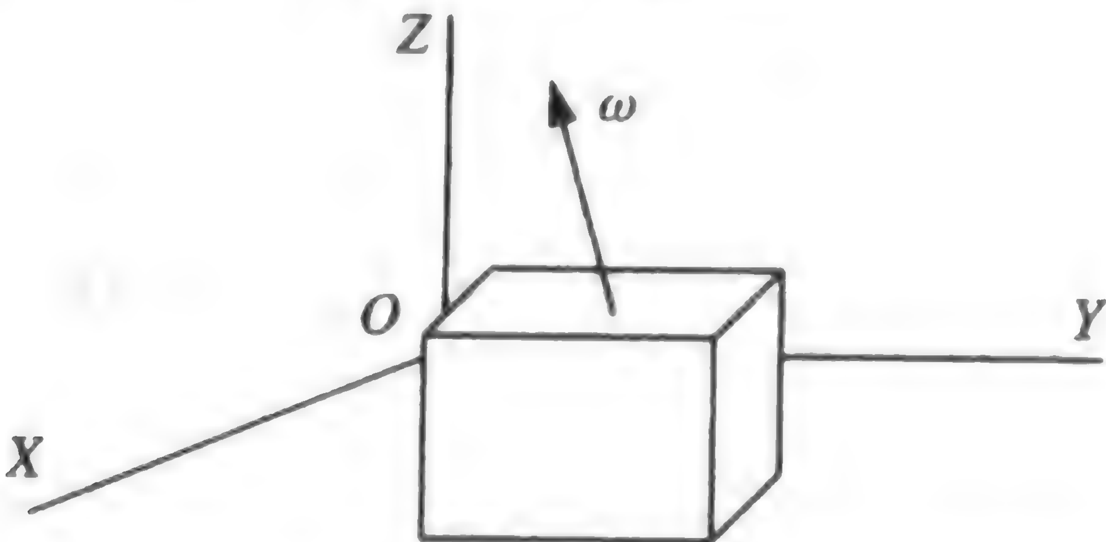


图 9.19　刚体在笛卡儿坐标轴中的转动

例 9.12　主惯性矩问题

现定义一组新的正交坐标轴,使其转动惯量矩阵呈对角线分布,这组坐标轴称为主轴,相应的惯性矩称为主惯性矩。式(9.25)的特征值即为主惯性矩,其相应的特征向量定义了主轴①。如下所述,这两个量由 eig 函数求得。注意,矩阵的特征值之和等于矩阵的迹。矩阵的迹为矩阵对角线元素之和,由 trace 函数求得。而且令:

$$[\boldsymbol{I}]=\begin{bmatrix}150 & 0 & -100\\ 0 & 250 & 0\\ -100 & 0 & 500\end{bmatrix}\text{kg}\cdot\text{m}^2 \tag{9.26}$$

程序如下:

```
Irot = [150,0, - 100;0,250,0; - 100,0,500];
[PrincipalDirections,PrincipalMoments] = eig(Irot)
TraceIrot = trace(Irot)
TracePM = trace(PrincipalMoments)
```

其执行结果如下:

```
PrincipalDirections =
        -0.9665        0       -0.2567
           0      1.0000          0
        -0.2567        0        0.9665
PrincipalMoments =
        123.4436        0        0
        0          250.0000      0
        0                 0 526.5564
TraceIrot =
 900
TracePM =
 900
```

尽管第一个特征向量只与第一个特征值有关,第二个特征向量只与第二个特征值有关,并可依次类推,但特征值(这里指主惯性矩)并没有一定的顺序,这点是 eig 函数所返回结果的特点。分析上述转动惯量矩阵发现,y 轴为一主轴,因此其中一个特征值等于 I_{yy}。主方向矩阵定义一个方向余弦矩阵,将(x,y,z)坐标轴转换为主轴。

刚体的稳定性

参照图 9.19 所示的刚体,设 $\boldsymbol{I}_1$,$\boldsymbol{I}_2$ 和 $\boldsymbol{I}_3$ 分别表示沿物体各个主轴方向定义的 x 轴、y 轴及 z 轴的第二质量惯性矩。即前例中的主惯性矩,ω_1,ω_2 及 ω_3 分别表示各个主轴的角速度,$\boldsymbol{M}_1$,$\boldsymbol{M}_2$ 及 $\boldsymbol{M}_3$ 分别表示各轴的外惯性矩。其运动方程即著名的欧拉(Euler)方程,具有以下形式②:

$$\begin{aligned}\boldsymbol{I}_1\dot{\omega}_1+(\boldsymbol{I}_3-\boldsymbol{I}_2)\omega_2\omega_3&=\boldsymbol{M}_1\\ \boldsymbol{I}_2\dot{\omega}_2+(\boldsymbol{I}_1-\boldsymbol{I}_3)\omega_3\omega_1&=\boldsymbol{M}_2\\ \boldsymbol{I}_3\dot{\omega}_3+(\boldsymbol{I}_2-\boldsymbol{I}_1)\omega_1\omega_2&=\boldsymbol{M}_3\end{aligned} \tag{9.27}$$

① 参见 D.T.Greenwood,出处同上。

② 参见 D.T.Greenwood,出处同上,第 392 页;F.C.Moon,出处同上,第 192 页。

其中：

$$\dot{\omega}_j = \frac{d\omega_j}{dt} \qquad j = 1,2,3$$

在自由振动情况下，即 $M_1 = M_2 = M_3 = 0$ 时，ω_i 不随时间而变化，其解有三种类型。这些解称做常数解，如下所示：

1. $(\omega_{10} \neq 0, \omega_{20} = 0, \omega_{30} = 0)$
2. $(\omega_{10} = 0, \omega_{20} \neq 0, \omega_{30} = 0)$
3. $(\omega_{10} = 0, \omega_{20} = 0, \omega_{30} \neq 0)$

每种解对应一个主轴方向的纯旋转动运动。这里只关心怎样求出三种振动的稳定位置。为此，令 $\xi_j(j = 1,2,3)$ 表示系统各轴上所加干扰，则有：

$$\begin{aligned} \omega_1(t) &= \omega_{10} + \xi_1(t) \\ \omega_2(t) &= \omega_{20} + \xi_2(t) \\ \omega_3(t) &= \omega_{30} + \xi_3(t) \end{aligned} \tag{9.28}$$

将式(9.28)代入式(9.27)，并假定干扰量数值很“小”，则可对式(9.27)线性化①，从而分析相关的特征值问题。其结果如下：

$$\begin{bmatrix} 0 & (I_3 - I_2)\omega_{30}/I_1 & (I_3 - I_2)\omega_{20}/I_1 \\ (I_1 - I_3)\omega_{30}/I_2 & 0 & (I_1 - I_3)\omega_{10}/I_2 \\ (I_2 - I_1)\omega_{20}/I_3 & (I_2 - I_1)\omega_{10}/I_3 & 0 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \lambda \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \tag{9.29}$$

在式(9.29)的三个特征值中，如果有一个或更多个具有正实部，则干扰值将增加，对应的振动将不稳定。因式(9.29)的迹为 0，所以特征值之和也将为 0。

例 9.13　刚体的稳定性

令刚体的 $I_1 = 150\ \text{kg}\cdot\text{m}^2$，$I_2 = 50\ \text{kg}\cdot\text{m}^2$，$I_3 = 300\ \text{kg}\cdot\text{m}^2$，分析三种常数解情况下每种解的稳定性。下面的程序用于求沿最大惯性轴、最小惯性轴及另一惯性轴加于旋转振动上的干扰量的特征值 λ：

```
I = [150,50,300];
omega10 = [1,0,0];
omega20 = [0,1,0];
omega30 = [0,0,1];
for i = 1:length(omega10)
  A = [0,(I(3) - I(2)) * omega30(i)/I(1),(I(3) - I(2)) * omega20(i)/I(1);...
      (I(1) - I(3)) * omega30(i)/I(2),0,(I(1) - I(3)) * omega10(i)/I(2);...
      (I(2) - I(1)) * omega20(i)/I(3),(I(2) - I(1)) * omega10(i)/I(3),0];
  fprintf(1,'\nCase %3d:Eigenvalues\n',i);
  lambda = eig(A)
  SumLambda = sum(lambda)
end
```

运行结果在 MATLAB 命令窗口中显示下列数据：

① 参见 A.H.Nayfeh 和 B.Balachandran，出处同上。

```
Case   1:Eigenvalues
lambda =
    1
  -1
    0
sum _ lambda =
  0
Case   2:Eigenvalues
lambda =
      0+0.7454i
      0-0.7454i
      0
sum _ lambda =
0
Case   3:Eigenvalues
lambda =
      0+2.2361i
      0-2.2361i
      0
sum _ lambda =
  0
```

对上述三种情况,每一种的特征值总有一个为0,而且特征值之和总为0。对第一种情况,其初始转动方向为沿惯性矩中间值的坐标轴方向,有一个特征值具有正实部,说明振动是不稳定的,对应的物理振动则是摇摆的振动。对第二种情况,其初始转动振动方向为沿惯性矩最小的坐标轴方向,由两个特征值组成一对纯虚数。对第三种情况,其初始旋转振动方向为沿惯性矩最大的坐标轴方向,由两个特征值组成一对纯虚数。在第二种和第三种情况下,因为没有一个特征值有正实部,所以系统所受的干扰不会增加,因此这两种情况下系统的振动是稳定的。如进一步用数值积分方法分析式(9.27)的后两种情况,可以发现振动轨迹为三维空间中的环形轨迹,用于定义三维空间的三个状态为 ω_1, ω_2 及 ω_3。这些轨迹位于 Poinsot 椭球面上①。

三自由度系统的振动模态及其固有频率

一系统如图 9.20 所示,x_1, x_2 和 x_3 为距离系统静止平衡位置的位移,系统的控制方程为:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{Bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{9.30}$$

其中,

$$\ddot{x}_j = \frac{\mathrm{d}^2 x_j}{\mathrm{d}t^2}$$

相应的特征值问题描述如下:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} = \lambda \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \tag{9.31}$$

其中,$\lambda = \omega^2$。

① 参见 D.T.Greenwood,出处同上,8.4 节。

例 9.14　三自由度系统的振动模态及固有频率

设 $k_1 = 100$ N/m, $k_2 = 50$ N/m, $m_1 = m_2 = m_3 = 100$ kg。求特征值及其相关特征向量的程序如下：

```
k=[100,-100,0;-100,150,-50;0,-50,50];
m=diag([100,100,100]);
[VibrationModes,Eigenvalues]=eig(k,m)
```

执行结果为：

```
VibrationModes =
  0.5774  0.5774  -0.5774
 -0.7887  0.5774  -0.2113
  0.2113  0.5774   0.7887
Eigenvalues =
 2.3660   0        0
   0      0.0000   0
   0      0        0.6340
```

观察图 9.20 所示系统，发现质量为 m_1，m_2 和 m_3 的小车两端并未受到弹簧的弹性恢复力，因此，该系统具有刚体模式。该模式中所有物体以相同方向移动相同的值。这一点在对应的振动模式中已体现出来，如振动模式矩阵中第 2 列元素所示。这种情况下弹簧既没有伸长也没有缩短，振动的特征值为 0。

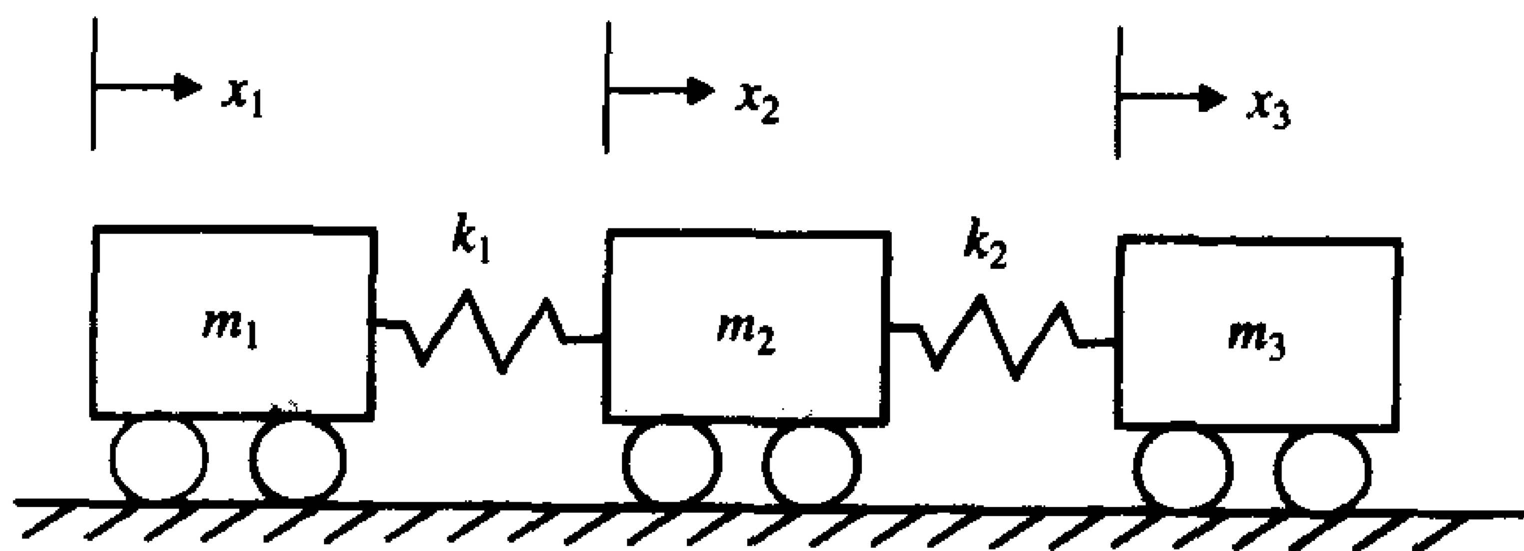


图 9.20　三自由度系统

如果一方阵有一零特征值，则矩阵的行列式为 0。为证实矩阵是否有零特征值，可求出矩阵的秩。矩阵的秩是矩阵中行列式不为 0 的最大子方阵的阶，可由函数 rank(**K**)求得。其中，**K** 为矩阵。

下列程序求式(9.30)所示的刚性矩阵是否有一零特征值：

```
k=[100,-100,0;-100,150,-50;0,-50,50];
rnk=rank(k);
[m n]=size(k);
disp(['Number of zero eigenvalues is 'num2str(m-rnk,2)])
```

执行结果产生以下输出：

```
Number of zero eigenvalues is 1
```

如果刚性矩阵的秩为 2，说明在这个(3×3)阶刚性矩阵中有一个(2×2)阶矩阵的行列式不为 0。

9.3.2 强迫振动及减振器

如图 9.21 所示,一受外力 $F(t)$作用的两自由度系统,其中质量 m_2、弹簧 k_2 及阻尼器 c_2 组成第二级系统,质量 m_1、弹簧 k_1 及阻尼器 c_1 组成主系统。第二级系统称做减振器,加在受外力作用的主系统之上以减弱主系统的振动。

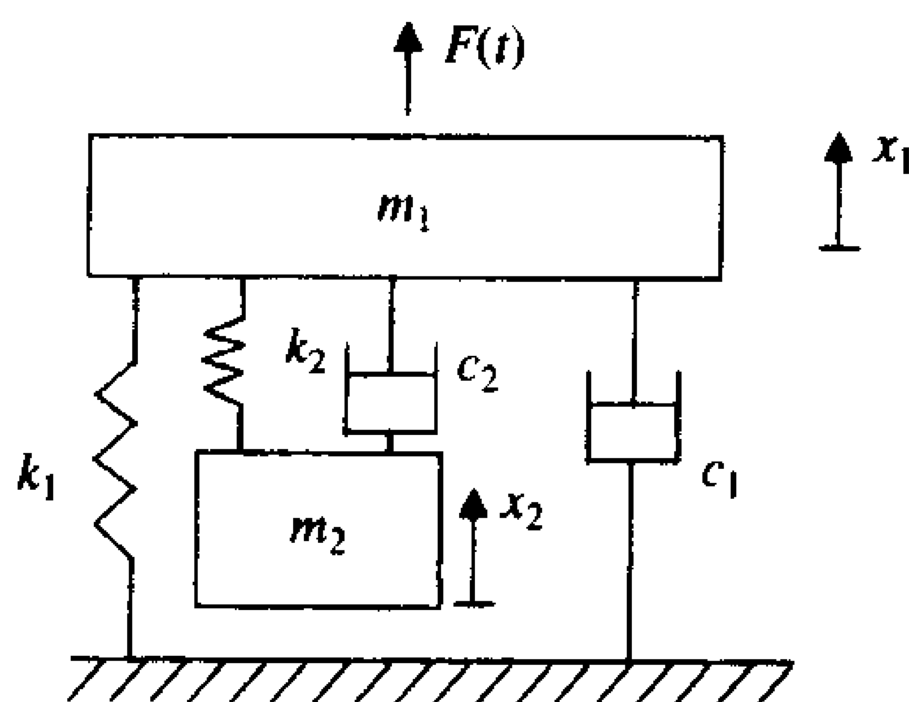


图 9.21 受外力 $F(t)$作用的两自由度系统

该系统的传递函数如下所示①:

$$\frac{\bar{x}_1(s)}{F(s)}=\frac{m_2s^2+c_2s+k_2}{D(s)}$$
$$\frac{\bar{x}_2(s)}{F(s)}=\frac{c_2s+k_2}{D(s)} \tag{9.32}$$

其中,

$$D(s)=m_1m_2s^4+[(c_1+c_2)m_2+c_2m_1]s^3+[(k_1+k_2)m_2+k_2m_1+c_1c_2]s^2+(k_1c_2+k_2c_1)s+k_1k_2 \tag{9.33}$$

由于这两个传递函数将多次应用,所以下面创建函数 **Transferab**:

```
function sys = Transferab(m,k,c)
N = {[m(2) c(2) k(2)];[c(2) k(2)]};
D = [m(1) * m(2)((c(1) + c(2)) * m(2) + c(2) * m(1))...
    ((k(1) + k(2)) * m(2) + k(2) * m(1) + c(1) * c(2))...
    (k(1) * c(2) + c(1) * k(2))k(1) * k(2)];
sy = tf(N,D);
```

例 9.15 对于两自由度系统的脉冲响应

在 $t=0$ 时,给质量为 m_1 的物体施加一单位脉冲,现在求质量为 m_1 和 m_2 的两物体的响应。假定:

$m_1=50$ kg	$k_1=200$ N/m	$c_1=10$ Ns/m
$m_2=10$ kg	$k_2=40$ N/m	$c_2=6$ Ns/m

程序如下:

```
m = [50,10];k = [200,40];c = [10,6];
[y,t] = impulse(Transferab(m,k,c),20);
figure(1)
```

① 参见 B. Balachandran 和 E. B. Magrab,出处同上。

```
plot(t,y(:,1),'k-',[0 20],[0 0],'k-')
ylabel('x_1(t)')
xlabel('t')
figure(2)
plot(t,y(:,2),'k-',[0 20],[0 0],'k-');
xlabel('t');
ylabel('x_2(t)');
```

程序执行结果如图 9.22 所示。从图中看出,两个物体在初始时瞬态运动是不同的。尽管脉冲加于物体 m_1 上,而物体 m_2 的响应幅值在初始时却大于 m_1 的响应幅值。经过大约 7 秒钟左右,两个物体的振动周期看起来相等。

通过将 impulse 函数换为 step 函数可以类似方法得到该系统的阶跃响应。

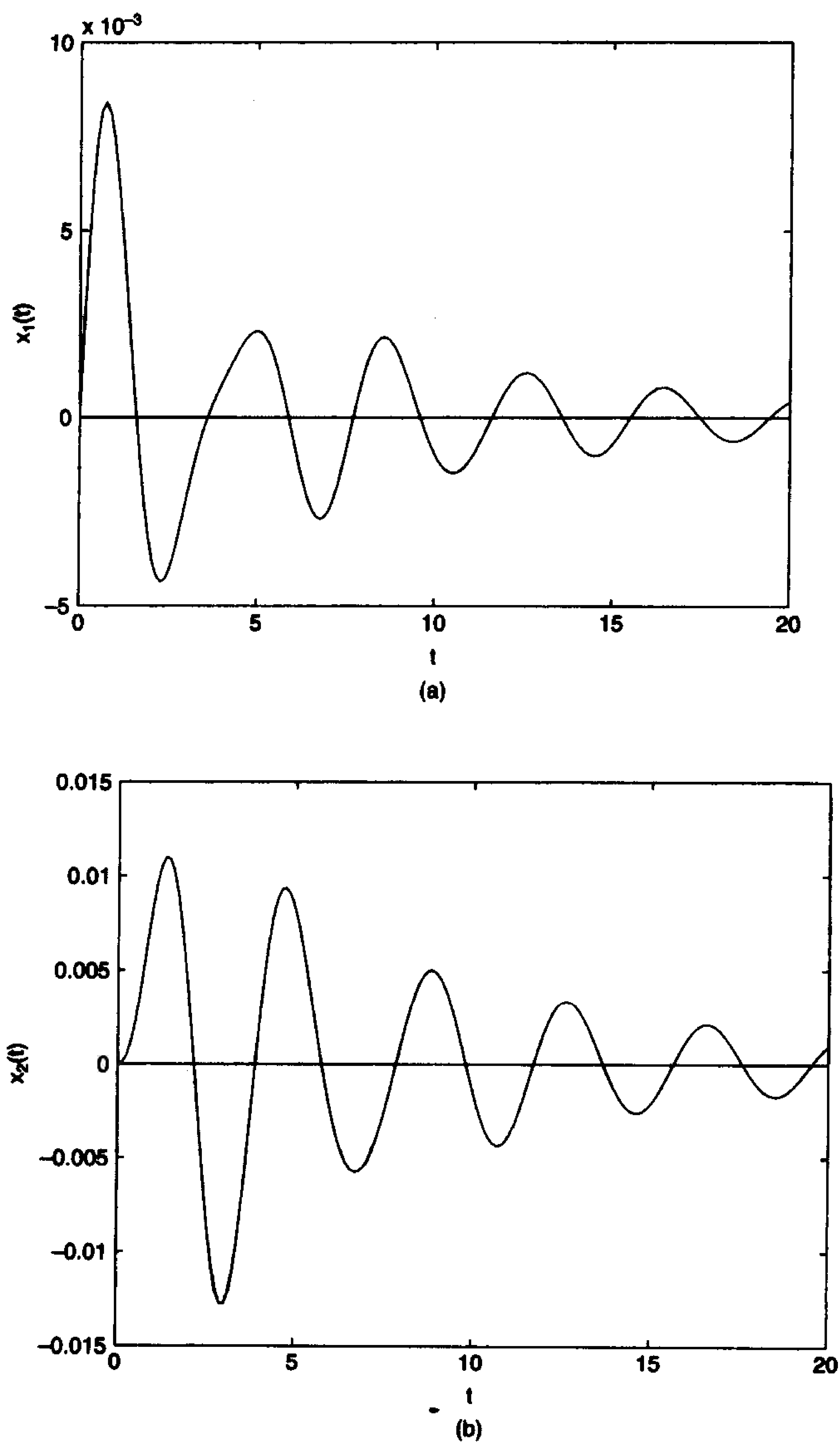


图 9.22 在 m_1 上加一脉冲时两自由度系统的脉冲响应:(a) m_1 的脉冲响应;(b) m_2 的脉冲响应

例 9.16　减振器

现在求 m_1 上施加一外力时，物体 m_1和 m_2位移响应的频率响应函数。这类分析方法用于减振器的设计。设计减振器的目的是减弱主要物体在干扰频率下的位移响应[①]。利用前例中相同的系统参数,程序代码为:

```
m = [50 10];k = [200 40];c = [10 6];
omega = linspace(0,4,300);
sys = tf([1],[m(1)c(1)k(1)]);
[mag phas] = bode(sys,omega);
plot(omega,mag(1,:),'--k');
hold on;
sys = Transferab(m,k,c);
[mag,phas] = bode(sys,omega);
plot(omega,mag(1,:),'k-');
xlabel('Excitation frequency(rad/s)');
ylabel('|x_1|');
text(2,1,0.045,'No vibration absorber');
text(0.3,0.02,'With vibration absorber');
```

程序执行结果如图 9.23 所示。由图上看出, m_1 的响应曲线在频率为 2 rad/s 附近衰减,这个值为主系统的无阻尼自然频率,即 m_2 的自然频率。有减振器的系统有两个自由度,相应的系统表现为有两个谐振频率。

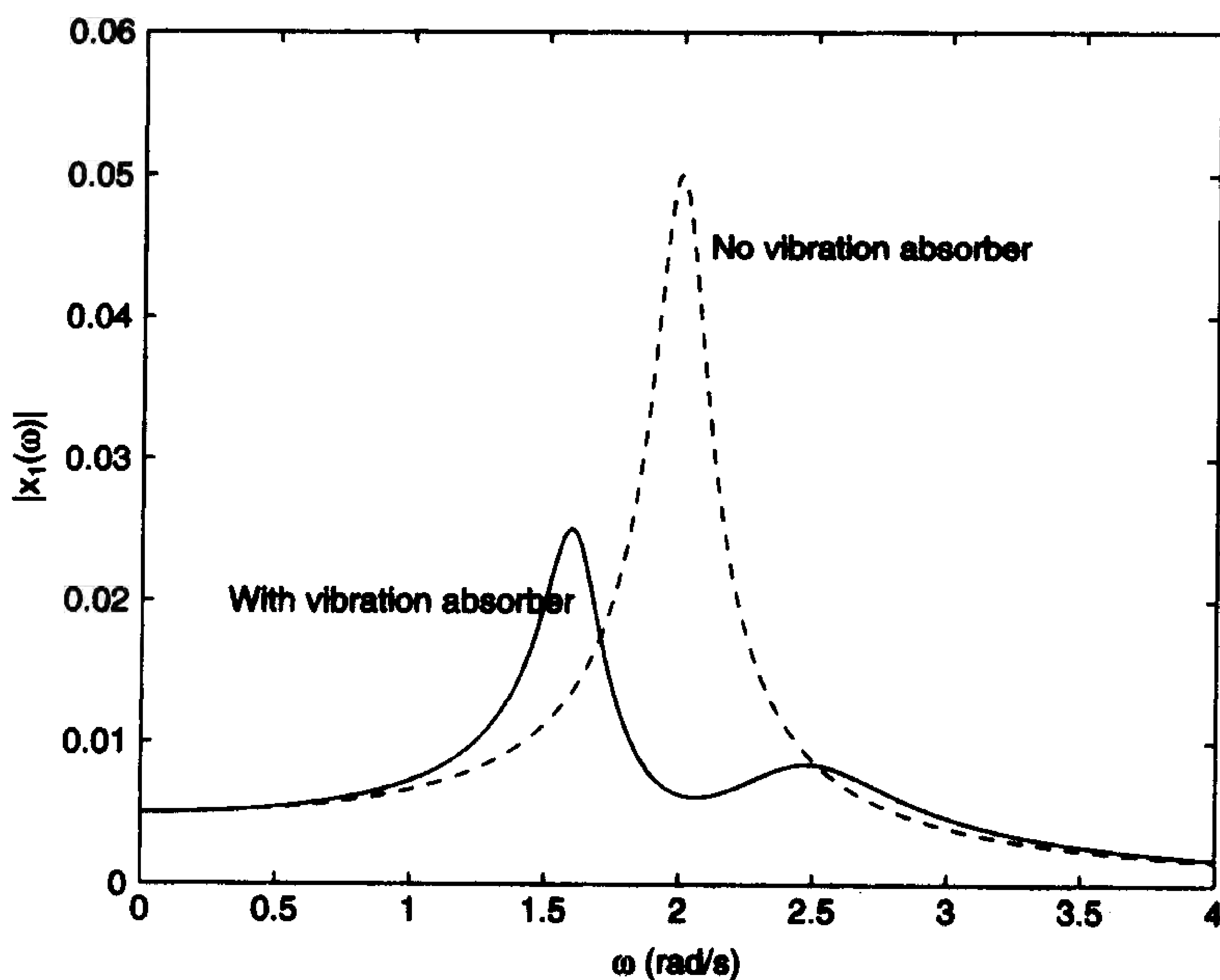


图 9.23　有减振器和没有减振器时第一质量系统的响应幅度

① 参见D.J.Inman,出处同上,5.3 节;S.S.Rao,出处同上,9.10 节;B.H.Tongue,出处同上,4.4 节;B.Balachandran和E.B,Magrab,出处同上,8.6 节。

例 9.17　减振器优化参数

从式(9.32)和式(9.33)开始。首先引入频率响应函数：

$$G_{11}(i\Omega)=\frac{\bar{x}_1(i\Omega)}{F(i\Omega)} \tag{9.34a}$$

并定义函数的幅度为：

$$H_{11}(\Omega)=|G_{11}(i\Omega)| \tag{9.34b}$$

其中，$\Omega=\omega/\omega_n$，$\omega_n=\sqrt{k_1/m_1}$。图 9.24 是一个减振器系统 $H_{11}(\Omega)$ 函数的典型图。为方便起见，引入以下无量纲参数：

$$\omega_r=\frac{\omega_{n2}}{\omega_{n1}}=\frac{1}{\sqrt{m_r}}\sqrt{\frac{k_2}{k_1}},m_r=\frac{m_2}{m_1}$$

$$\omega_{nj}^2=\frac{k_j}{m_j},2\zeta_j=\frac{c_j}{m_j\omega_{nj}}\quad j=1,2$$

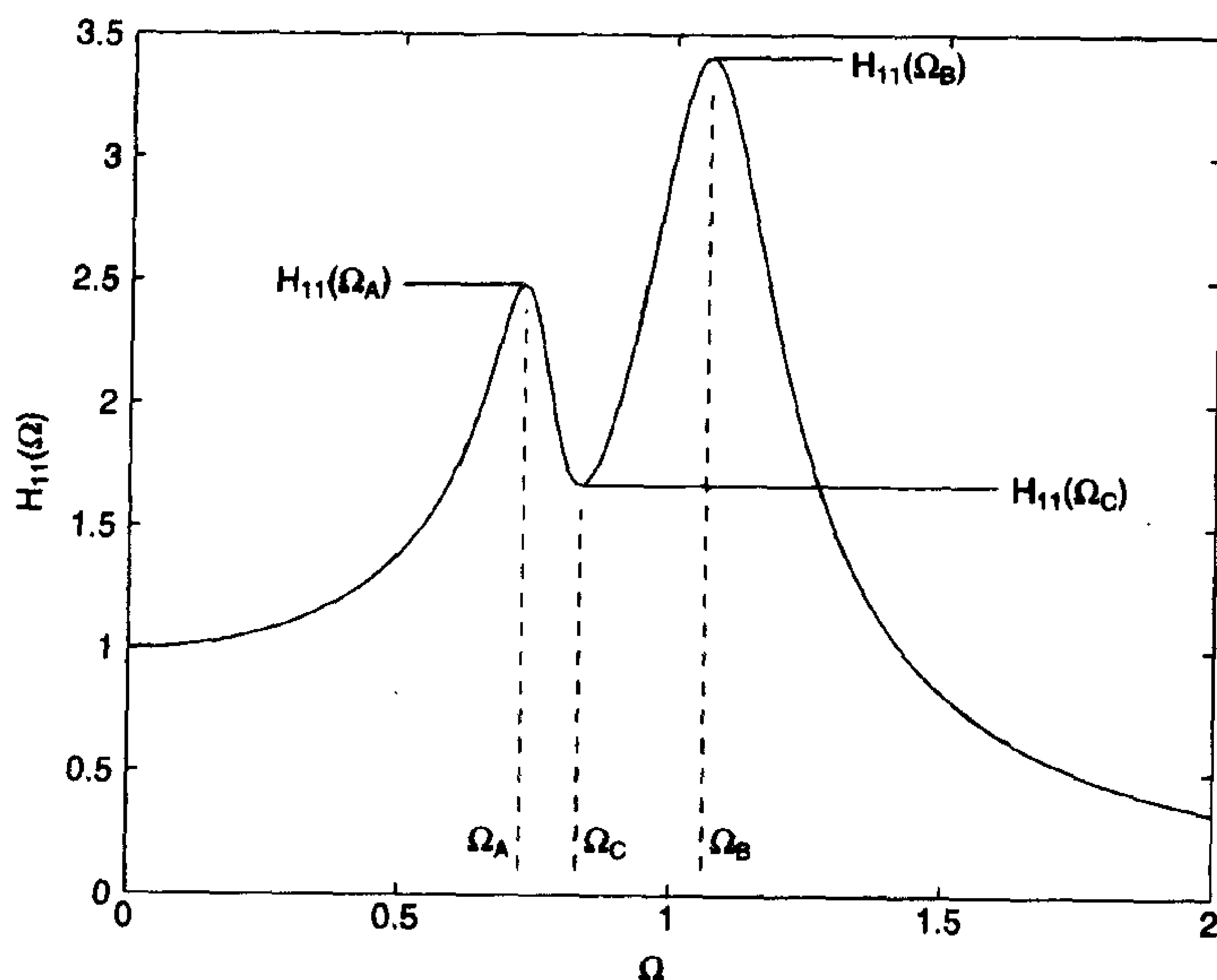


图 9.24　减振器系统典型幅度响应曲线

由式(9.32)的第一个式子和式(9.33)得到：

$$H_{11}(\Omega)=\left|\frac{E_2(i\Omega)}{k_1D_2(i\Omega)}\right|$$

其中，

$$E_2(j\Omega)=-\Omega^2+2\zeta_2\omega_rj\Omega+\omega_r^2$$

$$D_2(j\Omega)=\Omega^4-j[2\zeta_1+2\zeta_2\omega_rm_r+2\zeta_2\omega_r]\Omega^3-[1+m_r\omega_r^2+\omega_r^2+4\zeta_1\zeta_2\omega_r]\Omega^2+j[2\zeta_2\omega_r+2\zeta_1\omega_r^2]\Omega+\omega_r^2$$

现在的目标是求减振器的一组最优参数，形成一个包括 $\Omega=1$ 的操作区域，在该区域内主系统 m_1 大小随频率的变化最小。对应到图 9.24，目标就是找到减振器参数使图中 A 和 B 处峰值相等且尽可能小，而 A 和 B 之间的 C 点值尽可能接近 A 与 B 的值。换句话说，

要找的系统参数应同时使下列三个极大值最小化 $H_{11}(\Omega_A)$, $H_{11}(\Omega_B)$和 $1/H_{11}(\Omega_C)$。描述为:

$$\min_{\omega_r,\zeta_2}\{H_{11}(\Omega_A)\}$$
$$\min_{\omega_r,\zeta_2}\{H_{11}(\Omega_B)\}$$
$$\min_{\omega_r,\zeta_2}\{1/H_{11}(\Omega_C)\}$$
$$\text{subject to: } \omega_r > 0$$
$$\zeta_2 \geq 0 \tag{9.35}$$

$\zeta_1 = 0.1$ 且质量比 $m_r = 0.1, 0.2, 0.3, 0.4$ 时求最优值 $\zeta_{2,opt}$和 $\omega_{r,opt}$的脚本如下所示。对应这些最优值处的响应曲线如图 9.25 所示:

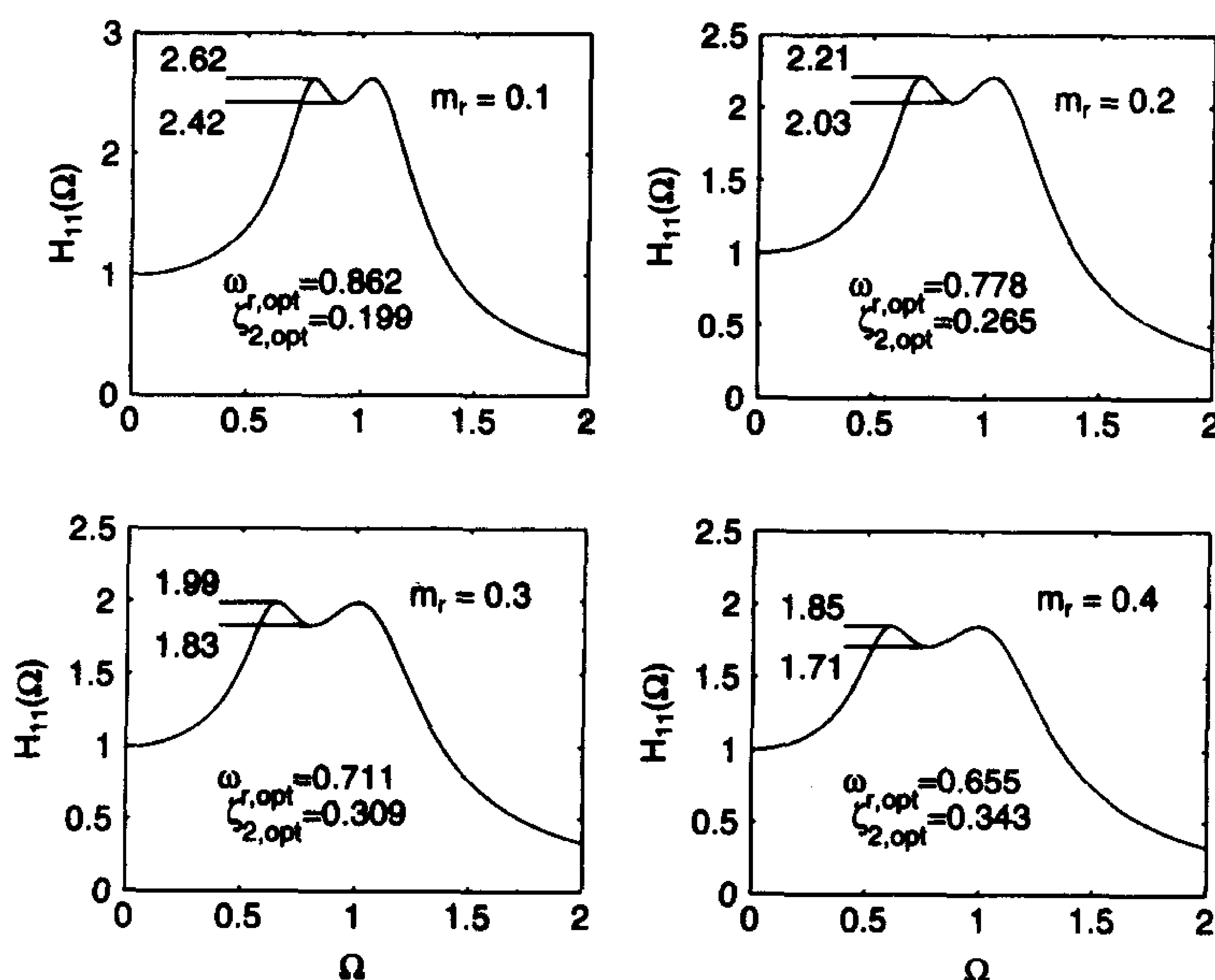


图 9.25　减振器系统的最优值及其在 $\zeta_1 = 0.1$ 时, m_1 的幅度响应曲线

```
function VibAbsorbOptPara
OM = linspace(0,2,100);
Lbnd = [0.1,0];
Ubnd = [2 1];
xo = [.8,0.35];
opt = optimset('Display','off');
mr = [.1.2.3.4];
z1 = [.1.1.1.1];
for k = 1:4
  subplot(2,2,k)
  [xopt,fopt] = fminimax (@objfun2dofHinconstr,xo,[],[],[],[],Lbnd,Ubnd,[],…opr,mr(k),z1
                  (k));
  plot(OM,Hal(OM,xopt(1),mr(k),z1(k),xopt(2)),'k-')
  hold on
  ax = axis;
  if k ~ = 1
```

```
    ax(4) = 2.5;
    axis(ax)
  end
  text(.2 * ax(2), .3 * ax(4), [' \ omega _{r,opt} = 'num2str(xopt(1),3)])
  text(.2 * ax(2), .2 * ax(4), [' \ zeta _{2,opt} = 'num2str(xopt(2),3)])
  text(1.3.8 * ax(4), ['m _ r = 'num2str(mr(k))])
  if k > 2
    xlabel(' \ Omega')
  end
    ylabel('H _{11}( \ Omega)')
    [z,xx] = objfun2doflinconstr([xopt(1),xopt(2)],mr(k),zl(k));
    hold on
    plot([xx(1),0.2 * ax(2)],[fopt(1)fopt(1)],'k-')
    text(.14 * ax(2),fopt(1),num2str(fopt(q),3))
    plot([xx(3),0.2 * ax(2)],[fopt(3)fopt(e)],'k-')
    text(.14 * ax(2),fopt(3),num2str(fopt(3),3))
  end

  function[z,xx] = objfun2doflinconstr(x,mr,zl)
  wr = x(1);
  z2 = x(2);
  opr = optimset('Display','off');
  al = 1 + (1 + mr) * wr^2;
  a2 = wr^2;
  O1 = sqrt(0.5 * (al - sqrt(ql^2-4 * a2)));
  O2 = sqrt(0.5 * (al + sqrt(ql^2-4 * a2)));
  [xl,fl] = fminsearch(@Min2dof,O1,opt,wr,mr,zl,z2);
  [x2,f2] = fminsearch(@Min2dof,O2,opt,wr,mr,zl,z2);
  [x3,z(3)] = fminsearch(@Hal,(O2 + O1)/2,opt,wr,mr,zl,z2);
  z(1) = 1/fl;
  z(2) = 1/f2;
  xx = [xl x2 x3];

  function h = Hal(Om,wr,mr,zl,z2)
  realpatr = Om.^4-(1 + mr * wr^2 + wr^2 + 4 * zl * z2 * wr) * Om.^2 + wr^2;
  imagpart = -2 * (zl + z2 * wr * mr + z2 * wr) * Om.^3 + 2 * (z2 * wr + zl * wr^2) * Om;
  nrealpart = wr^2-Om.^2;
  nimagpart = 2 * z2 * wr * Om;
  h = sqrt(nrealpart.^2 + nimagpart.^2)./sqrt(realpart.^2 + imagpart.^2);

  function m = Min2dof(Om,wr,mr,zl,z2)
  m = 1./Hal(Om,wr,mr,zl,z2);
```

9.4 细梁的振动[①]

9.4.1 等截面梁的自然频率与振动模态

一梁长度为 L，截面积为 A，惯性矩为 I，密度为 ρ，弹性模量为 E，受一动力 $F(x,t)$作用，其横向位移为 $w'(x,t)$时的无量纲振动控制方程为：

① 本节由 Edward B. Magrab 所写，部分内容摘自 B. Balachandran 和 E. B. Magrad, *Vibrations*, Brooks/Cole, Belmont, CA, 2002, 第9章。

$$\frac{\partial^4 w}{\partial \eta^4}+\frac{\partial^2 w}{\partial \tau^2}=F_o(\eta,\tau) \quad 0\leqslant\eta\leqslant 1 \quad \tau>0 \tag{9.36}$$

其中,$\eta=x/L$,$\tau=t/t_o$,$w(\eta,\tau)=w'(x/L,t/t_o)/L$,$F_o(\eta,\tau)=F(x/L,t/t_o)L^3/EI$,及:

$$t_o^2=\frac{\rho AL^4}{EI}$$

如图 9.26 所示的梁具有以下一般边界条件:

$\eta=0$

$$\frac{\partial^3 w}{\partial \eta^3}=-K_1 w$$

$$\frac{\partial^2 w}{\partial \eta^2}=B_1\frac{\partial w}{\partial \eta} \tag{9.37a}$$

$\eta=1$

$$\frac{\partial^3 w}{\partial \eta^3}=K_2 w+M\frac{\partial^2 w}{\partial \tau^2}$$

$$\frac{\partial^2 w}{\partial \eta^2}=-B_2\frac{\partial w}{\partial \eta}-J\frac{\partial^3 w}{\partial \eta\partial \tau^2} \tag{9.37b}$$

式(9.37)中,$M=M_o/m_o$,$J=J_o/j_o$,其中 M_o 为附加质量值。$m_o=\rho AL$ 为梁的质量,J_o 为附加质量的总惯性矩,$j_0=m_oL^2$,

$$K_j=\frac{k_jL^3}{EI} \quad 和 \quad B_j=\frac{k_{tj}L}{EI} \quad j=1,2$$

分别为弹簧的无量纲线性常数与扭转常数,k_j 为弹簧的线性常数,k_{tj}为扭转弹簧常数。

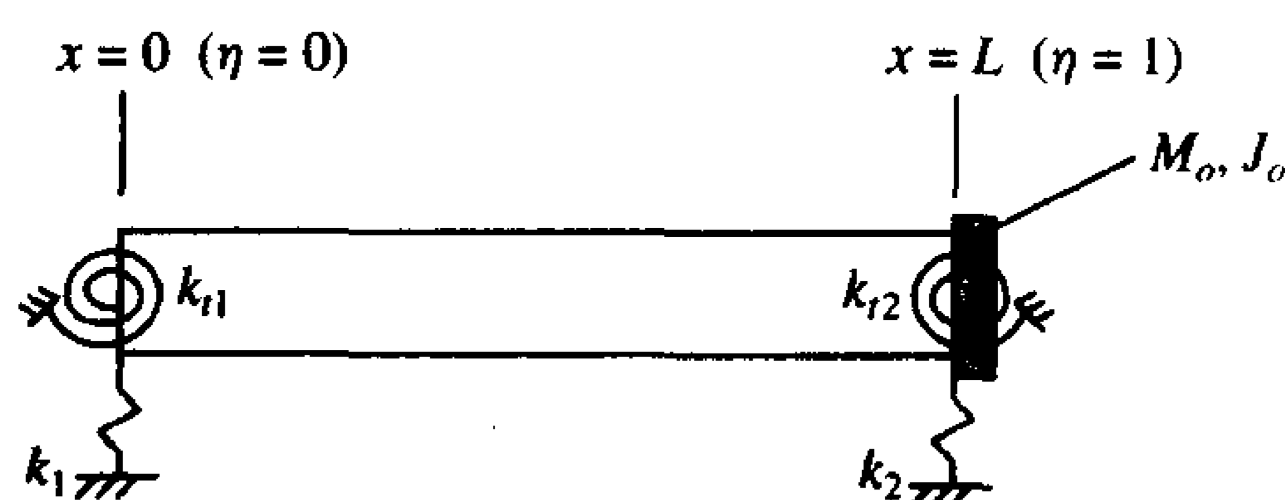


图 9.26 梁在左、右边界有弹簧元件,且右边界还有一个惯性元件

为求自然频率与振动模态,令式(9.36)中的 $F_o=0$,假定自由简谐振动函数形为:

$$w(\eta,\tau)=W(\eta)\cos(\Omega^2\tau) \tag{9.38}$$

其中,频率系数

$$\Omega=\sqrt{\omega t_o} \tag{9.29}$$

角频率 $\omega=2\pi f$。将式(9.38)代入式(9.37)得:

$\eta=0$

$$\frac{d^3 W}{d\eta^3}=-K_1 W$$

$$\frac{d^2 W}{d\eta^2}=B_1\frac{dW}{d\eta} \tag{9.39a}$$

$\eta=1$

$$\frac{d^3 W}{d\eta^3}=(K_2-M\Omega^4)W$$

$$\frac{d^2W}{d\eta^2} = (-B_2 + J\Omega^4)\frac{dW}{d\eta} \tag{9.39b}$$

令 $F_o = 0$，将式(9.38)代入式(9.36)求结果方程的一般解。利用式(9.39)给出的边界条件求下列特性方程，并由此计算自然频率系数 Ω_n 和相关振动模态 $W_n(\eta)$，特性方程如下：

$$\begin{aligned} & z_1[\cos\Omega_n \sinh\Omega_n + \sin\Omega_n \cosh\Omega_n] + z_2[\cos\Omega_n \sinh\Omega_n - \sin\Omega_n \cosh\Omega_n] \\ & -2z_3 \sin\Omega_n \sinh\Omega_n + z_4(\cos\Omega_n \cosh\Omega_n - 1) \\ & + z_5(\cos\Omega_n \cosh\Omega_n + 1) + 2z_6 \cos\Omega_n \cosh\Omega_n = 0 \end{aligned} \tag{9.40}$$

式中，

$$\begin{aligned} z_1 &= [b_{1n}b_{2n}(a_{1n} + a_{2n}) + (b_{1n} - b_{2n})] \\ z_2 &= [a_{1n}a_{2n}(b_{1n} - b_{2n}) - (a_{1n} + a_{2n})] \\ z_3 &= (a_{1n}a_{2n} + b_{1n}b_{2n}) \\ z_4 &= (1 - a_{1n}a_{2n}b_{1n}b_{2n}) \\ z_5 &= (a_{2n}b_{2n} - a_{1n}b_{1n}) \\ z_6 &= (a_{1n}b_{2n} - a_{2n}b_{1n}) \end{aligned} \tag{9.41}$$

和

$$\begin{aligned} a_{1n} &= \frac{K_1}{\Omega_n^3} \quad a_{2n} = \frac{1}{\Omega_n^3}(K_2 - M\Omega_n^4) \\ b_{1n} &= \frac{B^1}{\Omega_n} \quad b_{2n} = \frac{1}{\Omega_n}(-B_2 + J\Omega_n^4) \end{aligned} \tag{9.42}$$

振动模态与 b_{1n} 的范围有关：

情况 1：$0 \leqslant a_{jn} \leqslant \infty$，$0 \leqslant b_{2n} \leqslant \infty$ 及 $0 \leqslant b_{1n} < \infty$

振动模态为：

$$W_n(\eta) = C_n[Q(\Omega_n\eta) - a_{1n}T(\Omega_n\eta)] + R(\Omega_n\eta) + b_{1n}S(\Omega_n\eta) \tag{9.43a}$$

其中，无量纲系数 C_n 为：

$$C_n = \frac{a_{2n}R(\Omega_n) + (a_{2n}b_{1n} - 1)S(\Omega_n) - b_{1n}T(\Omega_n)}{R(\Omega_n) - (a_{1n} + a_{2n})Q(\Omega_n) + a_{1n}a_{2n}T(\Omega_n)} \tag{9.43b}$$

情况 2：$0 \leqslant a_{jn} \leqslant \infty$，$0 \leqslant b_{2n} \leqslant \infty$ 及 $b_{1n} \to \infty$（在 $\eta = 0$ 时的无限扭力强度）

振动模态为：

$$W_n(\eta) = C_n[Q(\Omega_n\eta) - a_{1n}T(\Omega_n\eta)] + S(\Omega_n\eta) \tag{9.44a}$$

其中，无量纲系数 C_n 为：

$$C_n = \frac{a_{2n}S(\Omega_n) - T(\Omega_n)}{R(\Omega_n) - (a_{1n} + a_{2n})Q(\Omega_n) + a_{1n}a_{2n}T(\Omega_n)} \tag{9.44b}$$

式(9.43)与式(9.44)中的各个不同函数定义如下：

$$\begin{aligned} Q(u) &= [\cos(u) + \cosh(u)]/2 \\ R(u) &= [\sin(u) + \sinh(u)]/2 \\ S(u) &= [\cosh(u) - \cos(u)]/2 \\ T(u) &= [\sinh(u) - \sin(u)]/2 \end{aligned}$$

这些表达式为边界条件的函数。选择边界条件可涉及大量特殊情况。获得这些特殊情况的方法是通过将参数 a_{1n}，a_{2n}，b_{1n}和b_{2n}设为 0 或∞，或一些中间值。表 9.3 中总结了许多边界条件，还有对应的 a_{1n}，a_{2n}，b_{1n}和 b_{2n}的值。采用表 9.3 中的极限值，式(9.40)将简化为表 9.4 所示的几种边界条件对应的关系式。

表 9.3　式(9.41)至式(9.44)的几种特殊情况的参数值

边界条件	$\eta=0$		$\eta=1$	
	a_{1n}	b_{1n}	a_{2n}	b_{2n}
简单支撑	∞	0	∞	0
夹紧状态	∞	∞	∞	∞
自由状态	0	0	0	0
自由状态有 M_o	—	—	$-M\Omega_n$	0
自由状态有 J_o	—	—	0	$J\Omega_n^3$
自由状态有 k_j	K_1/Ω_n^3	0	K_2/Ω_n^3	0
自由状态有 k_{tj}	0	B_1/Ω_n	0	$-B_2/\Omega_n$
简单支撑且有 $-k_{tj}$	∞	B_1/Ω_n	∞	$-B_2/\Omega_n$

表 9.4　式(9.40)的几种特殊情况

边界条件		式(9.40)变为
$\eta=0$	$\eta=1$	
简单支撑	简单支撑	$\sin(\Omega_n)=0$
夹紧状态	夹紧状态	$\cos(\Omega_n)\cosh(\Omega_n)-1=0$
夹紧状态	自由状态且 $M_o\neq 0$，$J_o=0$	$M\Omega_n[\cos(\Omega_n)\sinh(\Omega_n)-\sin(\Omega_n)\cosh(\Omega_n)]+\cos(\Omega_n)\cosh(\Omega_n)+1=0$
夹紧状态	自由状态	$\cos(\Omega_n)\cosh(\Omega_n)+1=0$
夹紧状态	简单支撑	$\tanh(\Omega_n)-\tan(\Omega_n)=0$

下面在式(9.39)定义的边界条件的任意组合下，用式(9.40)至式(9.44)求自然频率与模态函数。为此，首先创建函数，该函数使用符号工具箱中的相应函数求极限，然后根据所求得的符号表达式创建函数。函数名为 **BeamFreqMode**，如下所示。接着在该函数下面给出两个使用它的例子。在求解特性方程根的过程中，做了一些初级工作以求出合适的 fzero 的搜索区域。

```
function[Omeg,Mshape] = BeamFreqMode
global Bn1 Kn1 Bn2 Kn2 M J Nfreq BCtypeLeft BCtypeRight Neta
syms a1 a2 b1 b2 x K1 K2 B1 B2 Mm Jj et
z1 = b1 * b2 * (a1 + a2) + b1-b2;
z2 = a1 * a2 * (b1-b2)-a1-a2;
z3 = a1 * a2 + b1 * b2;
z4 = 1-a1 * a2 * b1 * b2;
z5 = a2 * b2-a1 * b1;
z6 = a1 * b2-a2 * b1;
chareqn = z1 * (cos(x) * sinh(x) + sin(x) * cosh(x))...
        + z2 * (cos(x) * sinh(x)-sin(x) * cosh(x))...
        -2 * z3 * sin(x) * sinh(x) + z4 * (cos(x) * cosh(x)-1)...
```

```
        + z5 * (cos(x) * cosh(x) + 1) + 2 * z6 * cos(x) * cosh(x);
Q = (cosh(x) + cos(x))/2;
S = (cosh(x) - cos(x))/2;
R = (sinh(x) + sin(x))/2;
T = (sinh(x) - sin(x))/2;
Cn = (a2 * R + (a2 * b1-1) * S-b1 * T)/(R-(a1 + a2) * Q + a1 * a2 * T);
Qo = (cosh(et * x) + cos(et * x))/2;
So = (cosh(et * x) - cos(et * x))/2;
Ro = (sinh(et * x) + sin(et * x))/2;
To = (sinh(et * x) - sin(et * x))/2;
Wn = Cn * (Qo - a1 * To) + Ro + b1 * So;
switch BCtypeLeft
  case 'clamped'
    chareqn = limit(limit(chareqn/a1,a1,inf)/b1,b1,inf);
    Cnb1 = (a2 * S-T)/(R-(a1 + a2) * Q + a1 * a2 * T);
    Wn = limit(Cnb1 * (Qo-a1 * To) + So,a1,inf);
  case 'hinged'
    chareqn = limit(limit(chareqn/a1,a1,inf),b1,B1/x);
    Wn = limit(limit(Wn,a1,inf),b1,B1/x);
  case 'free'
    chareqn = limit(limit(chareqn,b1,B1/x),a1,K1/x^3);
    Wn = limit(limit(Wn,b1,B1/x),a1,K1/x^3);
end
switch BCtypeRight
  case 'clamped'
    chareqn = limit(limit(chareqn/a2,a2,inf)/b2,b2,inf);
    Wn = limit(Wn,a2,inf);
    ul = 1.7 * pi;
  case 'hinged'
    chareqn = limit(limit(chareqn/a2,a2,inf),b2,-B2/x);
    Wn = limit(Wn,a2,inf);
    ul = 1.7 * pi;
case 'free'
    chareqn = limit(limit(chareqn,b2,-b2/x + Jj * x^3),a2,K2/x^3-Mm * x);
    Wn = limit(Wn,a2,K2/x^3-Mm * x);
    if length(BCtypeLeft) == 4
      ul = 1.7 * pi
    else
      ul = pi;
    end
end
charist = inline(vectorize(chareqn),'x','K1','K2','B1','B2','Mm','Jj');
modes = inline(vectorize(Wn),'x','et','K1',K2','B1','B2','Mn','Jj');
ll = 0.1 * pi;
eta = linspace(0,1,Neta);
opt = optimset('Display','off');
Omeg = zeros(1,Nfreq);
Mshape = zeros(Nfreq,Neta);
fork = 1:Nfreq
  Omeg(k) = fzero(charist,[ll,ul],opt,Kn1,Kn2,Bn1,Bn2,M,J);
  ll = pi * (Omeg(k)/pi + 0.1);
  ul = pi * (Omeg(k)/pi + 1.2);
```

```
    z = modes(Omeg(k),eta,Kn1,Kn2,Bn1,Bn2,M,J);
    Mshape(k,:) = z/max(z);
end
```

例 9.18　一端固定另一端装有铰链且有一扭力弹簧的梁的自然频率与振动模态

有一梁在 $\eta=0$ 处固定,在 $\eta=1$ 处装有铰链且带有一扭力弹簧,其无量纲扭力强度 $B_2=10$。下面求前 4 个自然频率与振动模态,并将结果绘成如图 9.27 所示的图形。程序如下:

```
global Bn1 Kn1 Bn2 Kn2 M J Nfreq BCtypeLeft BCtypeRight Neta
md = char('1st mode:','2nd mode:','3rd mode:','4th mode:');
Kn1 = 0;Bn1 = 0;
Kn2 = 0;Bn2 = 10;
M = 0;J = 0;
Nfreq = 4;Neta = 100;
BCtypeLeft = 'clamped';
BctypeRight = 'hinged';
[Omeg,Mshape] = BeamFreqMode;
eta = linspace(0,1,Neta);
for k = 1:Nfreq
  subplot(2,2,k)
  plot(eta,Mshape(k,:),'k-',[0,1],[0,0],'k--')
  title([md(k,:)' \ Omega _'num2str(k)' ∧ pi = 'num2str(Omeg(k)/pi)])
  axis off
end
```

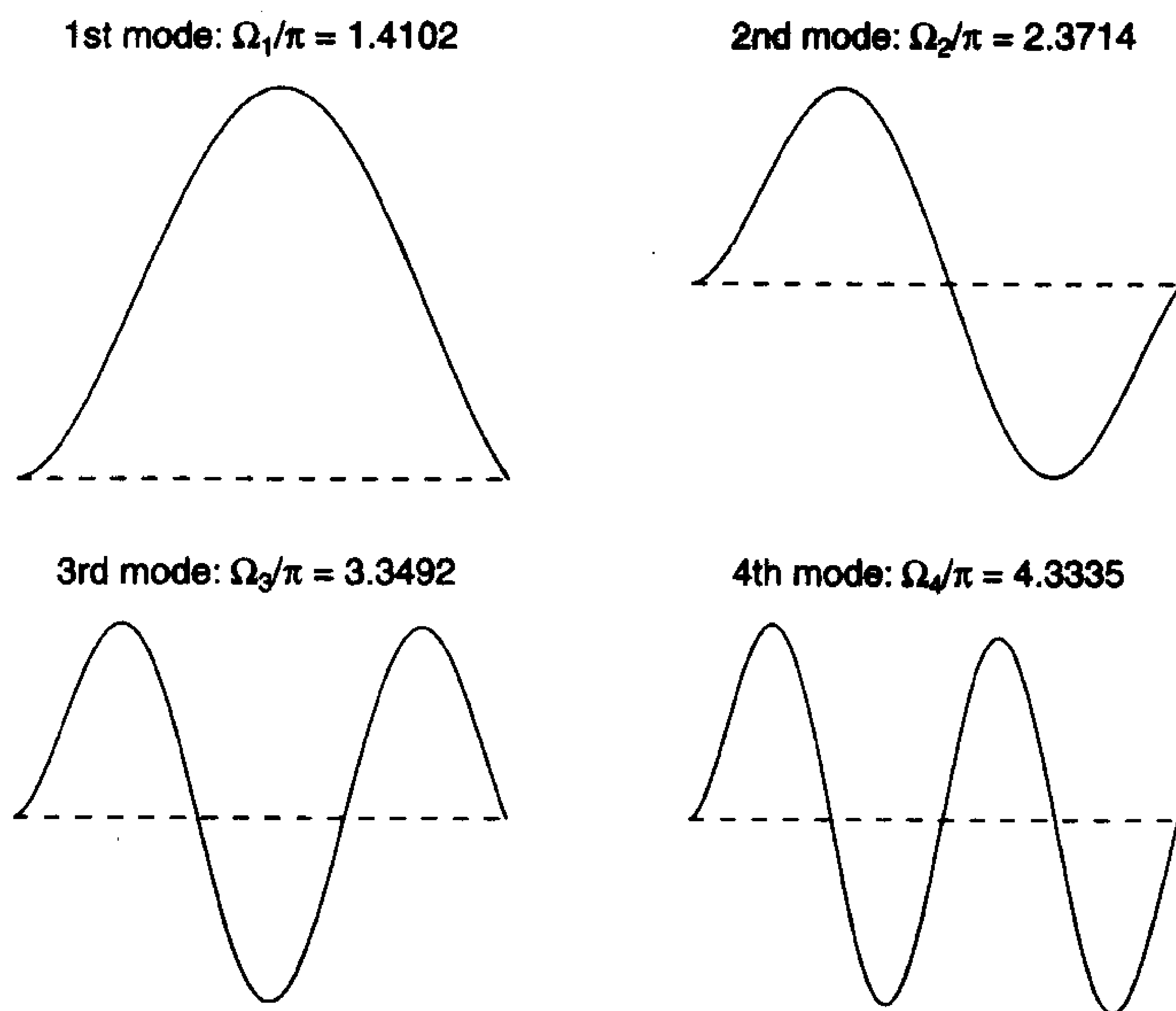


图 9.27　一端固定另一端装有铰链且 $B_2=10$ 的梁前 4 种振动模态

例 9.19　一端固定另一端自由且带有附加质量的梁的自然频率

有一梁在 $\eta=0$ 处固定,在 $\eta=1$ 处自由且有一附加质量,其无量纲质量比 M 在 $0.01\leqslant M\leqslant 100$ 范围内变化。下面在求作为 M 的函数的前两个自然频率,并将结果绘成如图 9.28 所示的图形。程序如下:

```
global Bn1 Kn1 Bn2 Kn2 M J Nfreq BCtypeLeft BCtypeRight Neta
Kn1 = 0; Bn1 = 0;
Kn2 = 0; Bn2 = 0;
J = 0;
Nfreq = 2; Neta = 100;
BCtypeLeft = 'clamped';
BCtypeRight = 'free';
Mm = logspace(-2,2,40);
LM = length(Mm);
omeg = zeros(LM,Nfreq);
for k = 1:LM
  M = Mm(k);
  Omeg(k,:) = BeamFreqMode/pi;
end
for k = 1:Nfreq
  semilogx(Mm,Omeg(:k),'k-')
  hold on
  text(1,0.8 * Omeg(1,k),[' \ Omega _'num2str(k)' ∧ pi])
end
xlabel('M = M _ o/m _ 0')
ylabel(' \ Omega _ n ∧ pi')
```

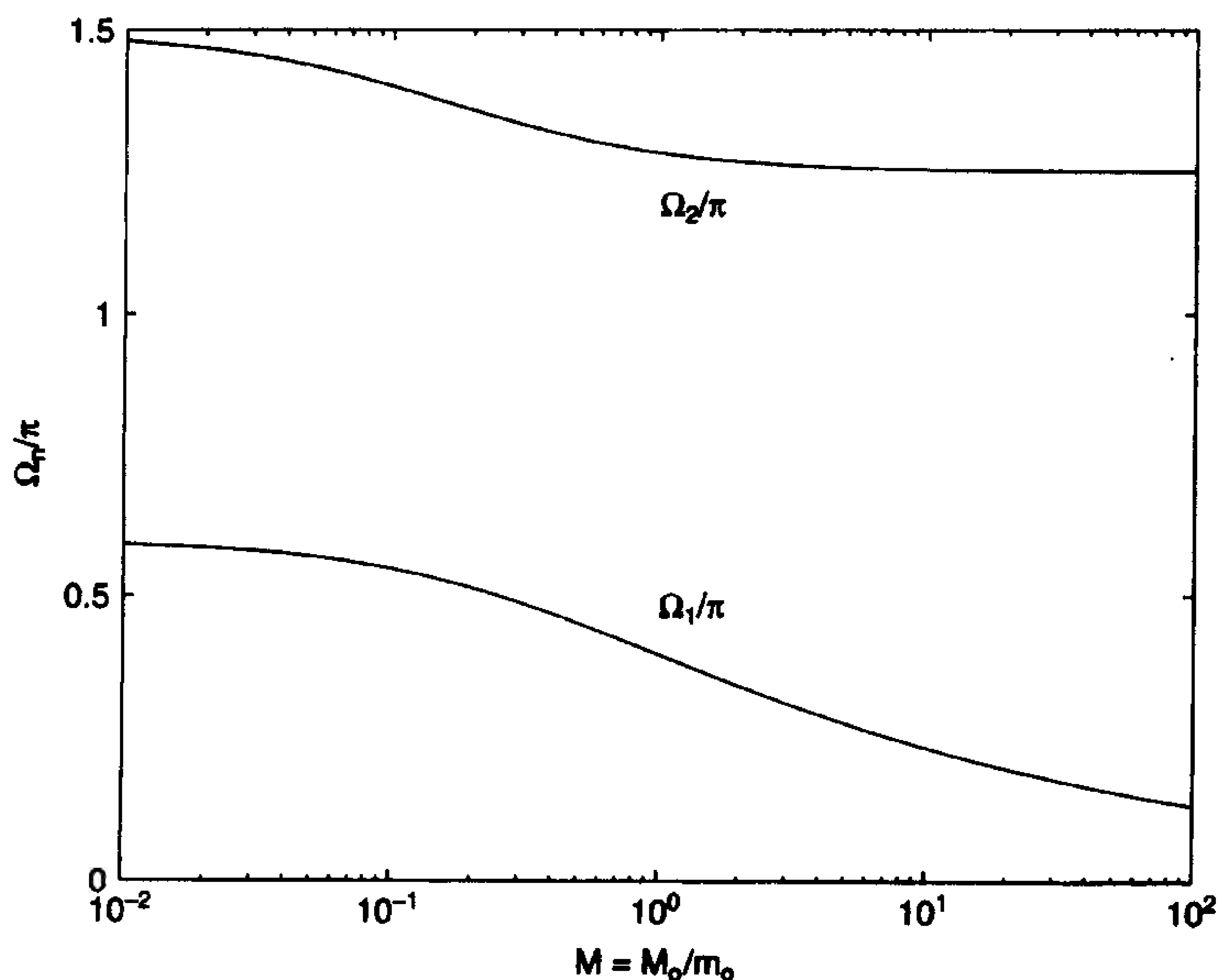


图9.28 一端固定另一端自由的梁前两个自然频率,其值为 M 的函数

9.4.2 梁的强迫振动

初始条件为0,约束条件为式(9.37)定义的边界条件时,式(9.36)的解为:

$$w(\eta,\tau) = \sum_{n=1}^{\infty} \frac{W_n(n)}{\Omega_n^2 N_n} \int_0^1 \int_0^{\tau} F_o(\eta,\tau') W_n(\eta) \sin[\Omega_m^2(\tau - \tau')] \mathrm{d}\tau' \mathrm{d}\eta \tag{9.45}$$

其中,$W_n(\eta)$为式(9.43a)或式(9.44a)确定的无量纲模态函数,

$$\Omega_n = \sqrt{\omega_n t_o}$$

$\omega_n = 2\pi f_n$ 是弧度自然频率，归一化常数为：

$$N_n = \int_0^1 W_n^2(\eta)\mathrm{d}\eta$$

下面举例说明其数值求解过程。

例 9.20 一端自由且带有附加质量的悬臂梁的脉冲响应

如梁受一脉冲 f_o 作用，且 $\eta = \xi(0<\xi<1)$ 时，有：

$$F_o(\eta,\tau) = f_o\delta(\eta-\xi)\delta(\tau)$$

则式(9.45)变为：

$$w(\eta,\tau) = f_o\sum_{n=1}^{\infty}\frac{W_n(\eta)W_n(\xi)}{\Omega_n^2 N_n}\sin(\Omega_n^2\tau)$$

$\xi=0.4, f_o=1$ 时画出在 $0.05\leqslant\tau\leqslant1.35$ 范围内 14 个时刻的梁的响应曲线。这里仅计算系列值的前 14 项，即仅考虑梁的前 14 个模态。下面的程序产生图 9.29 所示的结果：

```
global Bn1 Kn1 Bn2 Kn2 M J Nfreq BCtypeLeft BCtypeRight Neta
Kn1 = 0; Bn1 = 0;
Kn2 = 0; Bn2 = 0;
M = 0.2; J = 0;
Nfreq = 11; Neta = 100;
BCtypeLeft = 'clamped';
BCtypeRight = 'free';
[Omeg, Mshape] = BeamFreqMode;
eta = linspace(0,1,Neta);
tau = 0.05:0.1:1.35; xi = 0.4;
[v indx] = min(abs(eta-xi));
for k = 1:Nfreq
  Nn(k) = trapz(eta,Mshape(k,:),^2);
  Wnxi(k) = Mshape(k,indx);
end
C = Wnxi./Nn./Omeg.^2;
w = repmat(C,Neta,1). * Mshape' * sin(Omeg.^2' * tau);
Ntau = length(tau);
for k = 1:Ntau/2
  subplot(Ntau/2,2,2 * k-1)
  plot(eta,-w(:,k),'k-',[0,1],[0 0],'k--')
  axis([0 1 -0.4 0.4])
  text(0.05,0.2,[' \ tau = 'num2str(tau(k))])
  axis off
  subplot(Ntau/2,2,2 * k)
  plot(eta.-w(:,k + Ntau/2),'k-',[0 1],[0 0],'k--')
  axis([0 1 -0.4 0.4])
  text(0.05,0.2,[' \ tau = 'num2str(tau(k + Ntau/2))])
  axis off
end
```

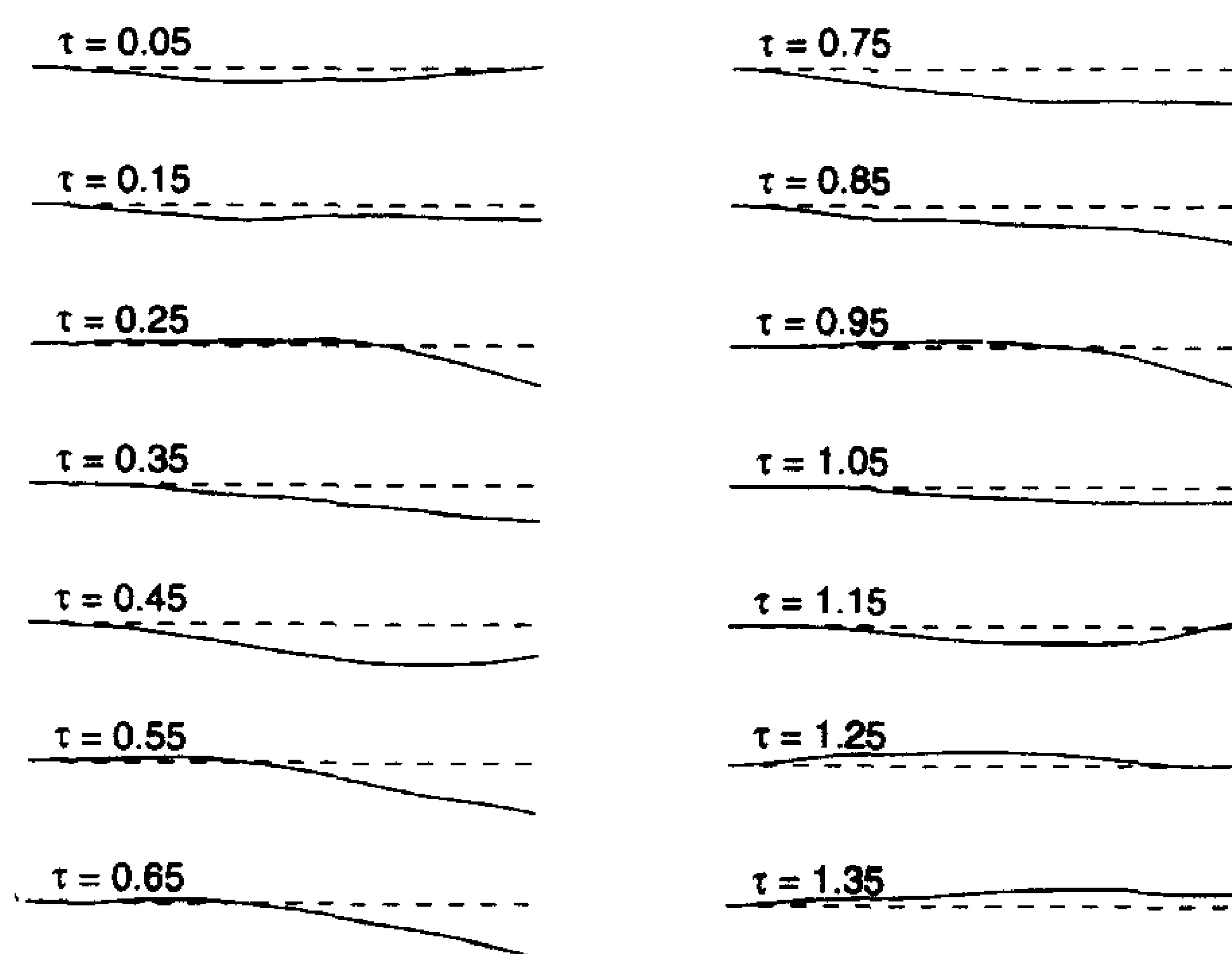


图 9.29　一端固定，一端自由且有 $M = M_o/m_o = 0.2$ 的质量的梁在 $\xi = 0.4$ 时的脉冲响应

9.4.3　变截面梁

图 9.30 所示为一变截面梁，称为双锥面梁在 z 方向做简谐振动的无量纲振动方程为：

$$\frac{d^2}{d\eta^2}\left[I(\eta)\frac{d^2 w}{d\eta^2}\right] - \Omega'^4 A(\eta) w = 0 \tag{9.46}$$

其中，

$$I(\eta) = [\alpha + (1-\alpha)\eta]^3[\beta + (1-\beta)\eta^{n-1}]$$
$$A(\eta) = [\alpha + (1-\alpha)\eta][\beta + (1-\beta)\eta^{n-1}]$$
$$\Omega'^4 = \omega^2 L^4 / c_b^2 r_0^2$$
$$c_b^2 = E/\rho$$

r_0 为 $\eta = 1$ 端截面的回转半径，$\alpha = h_1/h_0 \leqslant 1$ 为锥深比，$\beta = b_1/b_0 \leqslant 1$ 为锥的厚度比值，n 为大于 0 的常数，这种梁称做双锥面梁。$\beta = 0$ 时梁的 $\eta = 0$ 端在 xy 平面锥面缩为一点；$\alpha = 0$ 时，在 $\eta = 0$ 端锥面将沿 xz 平面缩小为一点；$\beta = \alpha = 1$ 时得到一等截面梁。当锥的厚度比和深度比与 η 成线性比例关系时，$n = 2$；当锥的厚度为常数时（$\beta = 1$），$n = 1$；当厚度比与深度比相等时，$\beta = \alpha$。

下面缩小讨论范围，只讨论 $\beta = \alpha$ 及 $n = 2$ 时的双锥面梁，则式(9.46)写为：

$$\frac{d^2}{d\varphi^2}\left[\varphi^4 \frac{d^2 W}{d\varphi^2}\right] - \lambda^4 \varphi^2 W = 0 \tag{9.47}$$

其中，

$$\varphi = [\alpha + (1-\alpha)\eta]$$
$$\lambda = \Omega'/(1-\alpha)$$

式(9.47)的解为：

$$W(\varphi) = \varphi^{-1}\left[AJ_2(2\lambda\sqrt{\varphi}) + BY_2(2\lambda\sqrt{\varphi}) + CI_2(2\lambda\sqrt{\varphi}) + DK_2(2\lambda\sqrt{\varphi})\right] \tag{9.48}$$

其中，$J_2(z)$，$Y_2(z)$，$I_2(z)$，$K_2(z)$分别为一类和二类 2 阶贝塞尔函数及修改过的 2 阶贝塞尔函数。

对式(9.39)用 $d\varphi/d\eta = (1-\alpha)$ 可求得边界条件。

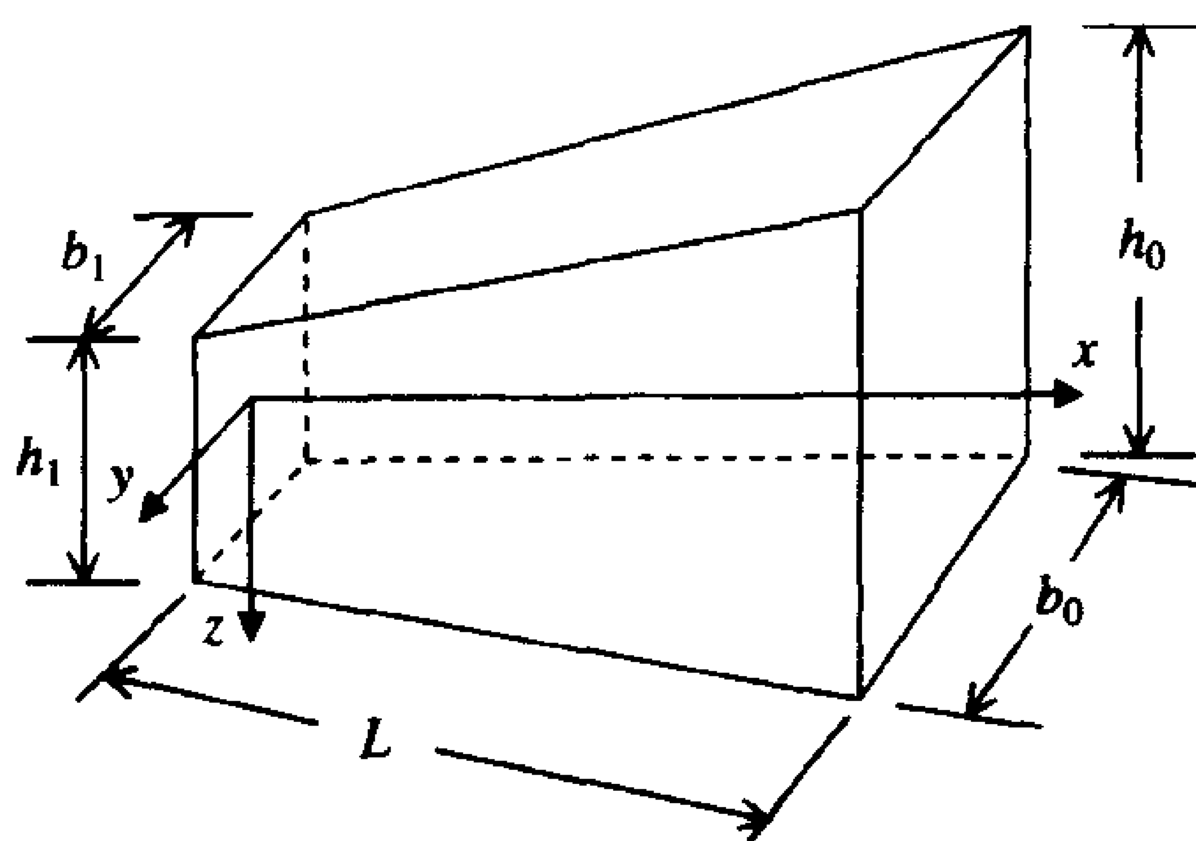


图 9.30　锥形梁的参数

例 9.21　双锥面悬臂梁的最低固有振动频率

假定一悬臂梁 $\eta=1$ 端固定，$\eta=0$ 端自由，现求其 α 在一段范围内取值时的三个最低固有振动频率系数。边界条件为：

$\eta=1(\varphi=1)$

$$W=0 \quad \text{和} \quad \frac{\mathrm{d}w}{\mathrm{d}\varphi}=0 \tag{9.49a}$$

$\eta=0(\varphi=\alpha)$

$$\frac{\mathrm{d}^2 w}{\mathrm{d}\varphi^2}=0 \quad \text{和} \quad \frac{\mathrm{d}^3 W}{\mathrm{d}\varphi^3}=0 \tag{9.49b}$$

将式(9.48)代入式(9.49)，得到下式，从而求得固有振动频率系数 λ_n：

$$\begin{vmatrix} J_5(2\lambda\sqrt{\alpha}) & Y_5(2\lambda\sqrt{\alpha}) & -I_5(2\lambda\sqrt{\alpha}) & K_5(2\lambda\sqrt{\alpha}) \\ J_4(2\lambda\sqrt{\alpha}) & Y_4(2\lambda\sqrt{\alpha}) & I_4(2\lambda\sqrt{\alpha}) & K_4(2\lambda\sqrt{\alpha}) \\ J_2(2\lambda) & Y_2(2\lambda) & I_2(2\lambda) & K_2(2\lambda) \\ J_3(2\lambda) & Y_3(2\lambda) & -I_3(2\lambda) & K_3(2\lambda) \end{vmatrix}=0 \tag{9.50}$$

为求得式(9.50)，用到下列等式：

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}J_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}J_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}Y_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}Y_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}I_n(2\lambda\sqrt{\varphi})]=\lambda\varphi^{-(n+1)/2}I_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}K_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}K_{n+1}(2\lambda\sqrt{\varphi})$$

当 $\beta=\alpha=1$ 时，可通过执行适用于悬臂梁情况的函数 **BeamFreqMode** 得到 λ_1 的值。结果为 $\lambda_1=\Omega'_1=1.8751$。下面程序中函数 **TaperedBeam** 表示式(9.50)，执行程度得图 9.31。

```
function TaperedBeamFreq
a=logspace(-1,0,20);
opt=optimset('Display','off');
for k=1:length(a)-1
```

```
    b(k) = fzero(@TaperedBeam,[1.5/(1-a(k)),4/(1-a(k))],opt,a(k)) * (1-a(k));
end
b(length(a)) = 1.8751;
semilogx(a,b,'k-')
xlabel('\alpha')
ylabel('\Omega\prime_1')
axis([.1 1 1.8 2.8])

function r = TaperedBeam(x,alpha)
a1 = 2 * x;
a2 = a1 * sqrt(alpha);
r = det([besselj(5,a2)bessely(5,a2) - besseli(5,a2)besselk(5,a2);...
        besselj(4,a2)bessely(4,a2)besseli(4,a2)besselk(4,a2);...
        besselj(2,a1)bessely(2,a1)besseli(2,a1)besselk(2,a1);...
        besselj(3,a1)bessely(3,a1) - besseli(3,a1)besselk(3,a1)]);
```

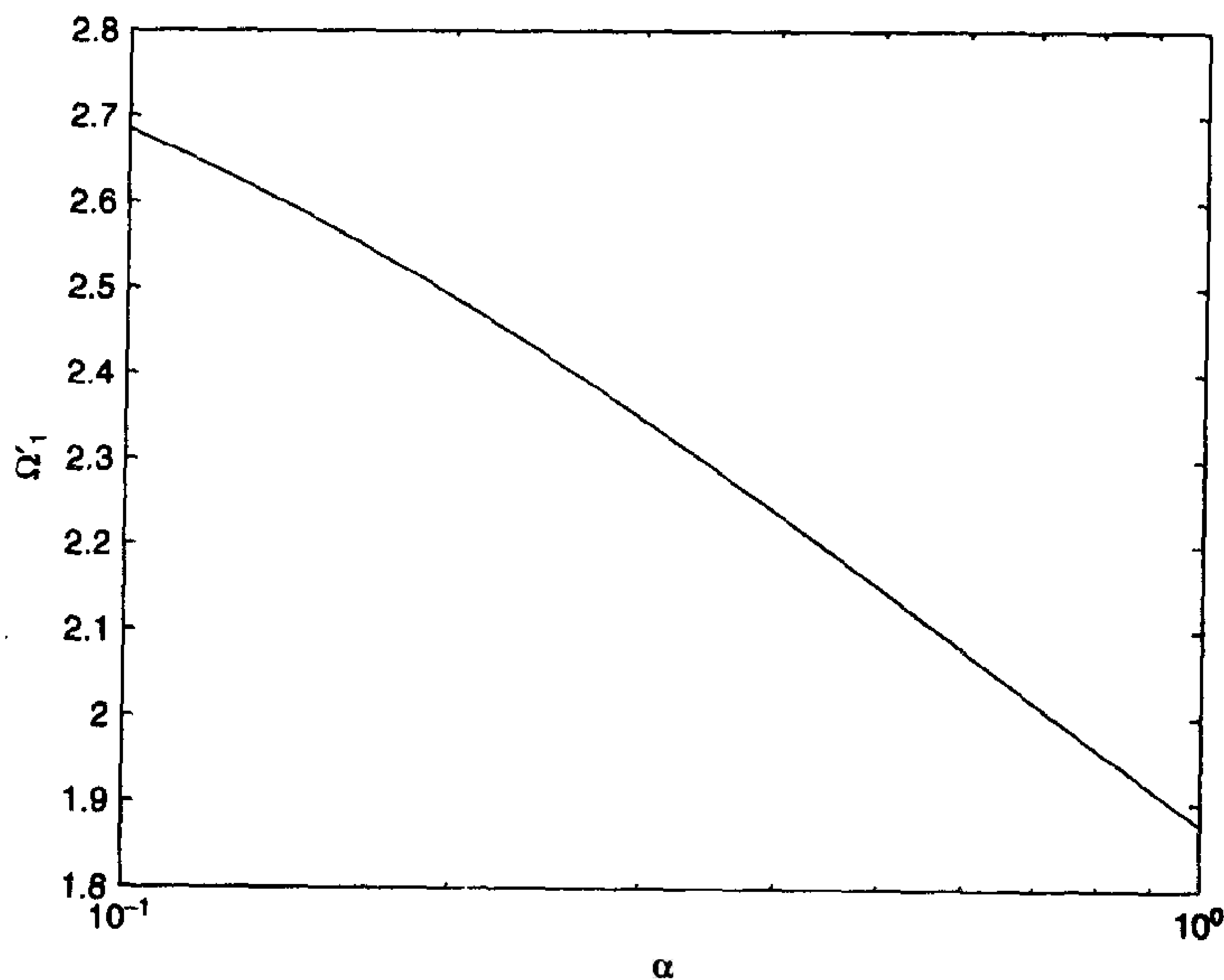


图 9.31　双锥面悬臂梁的第一个自然频率系数

9.4.4　有附加集中质量的梁

在 $\eta=\xi$ 处有附加集中质量 M'_0 的梁，做自由简谐振动的无量纲振动方程为：

$$\frac{d^4W}{d\eta^4}-\Omega^4\left[1+\frac{M'_0}{m_0}\delta(\eta-\xi)\right]W=0 \tag{9.51}$$

其中，$W\equiv W(\eta)$，边界条件由式(9.39)给出。式(9.51)的一般解为：

$$W(\eta)/W(\xi)=AQ(\Omega\eta)+BR(\Omega\eta)+CS(\Omega\eta)+DT(\Omega\eta)+\frac{M'_0}{m_0}\Omega T(\Omega(\eta-\xi))u(\eta-\xi) \tag{9.52}$$

其中，$u(\eta-\xi)$为单位阶跃函数：频率系数 Ω_n 由式(9.52)求出，取估计值 $\eta=\xi$。因 $T(0)=0$，所以有：

$$1=AQ(\Omega_n\xi)+BR(\Omega_n\xi)+CS(\Omega_n\xi)+DT(\Omega_n\xi) \tag{9.53}$$

将式(9.39)的边界条件代入式(9.52)求得常数 $A,\cdots,D$ 的值。对应的模态为：

$$w_n(\eta)=A_nQ(\Omega_n\eta)+B_nR(\Omega_n\eta)+C_nS(\Omega_n\eta)+D_nT(\Omega_n\eta)$$
$$+\frac{M'_0}{m_0}\Omega_nT(\Omega_n(\eta-\xi))u(\eta-\xi) \tag{9.54}$$

A_n,B_n,C_n 与 D_n 分别是 A,B,C 与 D 在 $\Omega=\Omega_n$ 时的值。下面举例说明这些结果。

例 9.22 考虑带质量简支梁的固有频率和模态

一两端简支梁，其质量所在位置 ξ 从 0.05 变到 0.5，M'_0/m_0 有三种值 0.1,1,10。下面求解梁的前三个固有频率系数，并画出图形。由于梁两端边界条件相同，所以 Ω_n 关于 $\xi=0.5$ 对称。另外，在图中画出了各种模态，并将其与两端简单支撑无附加质量的梁的模态进行了比较。

首先，边界条件为：

$$W(0)=\frac{d^2W(0)}{d\eta^2}=0$$
$$W(1)=\frac{d^2W(1)}{d\eta^2}=0 \tag{9.55}$$

将式(9.52)代入式(9.54)，发现 $A=C=0$，且有：

$$B=\frac{M'_0\Omega}{G_0m_0}[T(\Omega)R(\Omega\varepsilon)-R(\Omega)T(\Omega\varepsilon)]$$
$$D=\frac{M'_0\Omega}{G_0m_0}[T(\Omega)T(\Omega\varepsilon)-R(\Omega)R(\Omega\varepsilon)] \tag{9.56}$$
$$G_0=R^2(\Omega)-T^2(\Omega)$$

其中，$\varepsilon=1-\xi$，则式(9.53)变为：

$$R^2(\Omega_n)-T^2(\Omega_n)-\frac{M'_0}{m_0}\Omega_n\{R(\Omega_n\xi)[T(\Omega_n)R(\Omega_n\varepsilon)-R(\Omega_n)T(\Omega_n\varepsilon)]$$
$$+T(\Omega_n\xi)[T(\Omega_n)T(\Omega_n\varepsilon)-R(\Omega_n)R(\Omega_n\varepsilon)]\}=0 \tag{9.57}$$

得到的模态为：

$$W_n(u)=B_nV(\Omega_nu)+D_nT(\Omega_nu)+\frac{M'_0}{m_0}\Omega_nT(\Omega_n(u-\xi))H(u-\xi) \tag{9.58}$$

其中，B_n 与 D_n 在 $\Omega=\Omega_n$ 时由式(9.54)给出。

对无附加质量的两端简支梁，采用表 9.4 所列出的边界条件。

$$\Omega_n=n\pi$$
$$W_n(\eta)=\sin(n\pi\eta)$$

下面的主函数有两个子函数，第一个子函数为 **BeamMassCharEqn**，表示式(9.57)的特征方程；第二个子函数 **BeamMassMode** 计算式(9.58)的模态。以下程序执行时产生图 9.32 和图 9.33：

```
function BeamWithMass
Mom = [.1,1,10];
opt = optimset('Display','off');
xi = linspace(0.05,0.5,15);
L1 = [0.4,1.2,2.3] * pi;
U1 = [1.1,2.1,3.1] * pi;
for m = 1:length(Mom)
  for k = 1:length(xi)
```

```
      for n = 1:3
        coeff(m,k,n) = fzero(@BeamMassCharEqn,[L1(n),U1(n)],opt,xi(k),...
          Mom(m))/pi;
      end
    end
  end
  lab = char('first','second','third');
  figure(1)
  for kk = 1:3
    plot(xi,coeff(1,:,kk),'k-',xi,coeff(2,:,kk),'k--',xi,coeff(3,:,kk),'k-.')
    hold on
    text(.4,kk+.1,lab(kk,:))
  end
  axis([0.05,0.5,0,3.5])
  xlabel('Mass position, \xi')
  ylabel('\Omega_n/\pi')
  legend(['M\prime_0/m_0='num2str(MoM(1))],...
         ['M\prime_0/m_0='num2str(Mom(2))],...
         ['M\prime_0/m_0='num2str(Mom(3))],'Location','SouthWest')
  figure(2)
  sig = 1;
  lab = char('First mode','Second mode','Third mode');
  for k = 1:3
    if k == 3
      sig = -1;
    end
    for kk = 1:3
      subplot(3,3,3*(k-1)+kk)
      [shape,u] = BeamMassMode(coeff(kk,2,k)*pi,0,4);
      plot(u,shape,'k-',u,sig*sin(k*u*pi),'k--',[0 1],[0 0],'k-')
      axis([0 1 -1 1])
      axis off
      if k == 1
        title(['M\prime_0/m_0='num2str'(Mom(kk))])
      end
      fi kk == 1
        text(-.15,-.5,lab(k,:),'rotation',90)
      end
    end
  end

function Omn = BeamMassCharEqn(Om,xi,Mom)
ep = 1-xi;
R = inline('(sinh(x)+sin(x))/2','x');
T = inline('(sinh(x)-sin(x))/2','x');
p1 = R(Om)^2-T(Om)^2;
p2 = R(Om*xi)*(T(Om)*R(Om*ep)-R(Om)*T(Om*ep));
p3 = T(Om*xi)*(T(Om)*T(Om*ep)-R(Om)*R(Om*ep));
Omn = p1-Mom*Om*(p2+p3);

function[modeshape,etat] = BeamMassMode(Om,xi)
ep = 1-xi;
R = inline('(sinh(x)+sin(x))/2','x');
T = inline('(sinh(x)-sin(x))/2','x');
```

```
p1 = R(Om)^2 - T(Om)^2;
P2 = CT(Om) * R(Om * ep) - R(Om) * T(Om * ep))/p1;
p3 = CT(Om) * T(Om * ep) - R(Om) * R(Om * ep))/p1;
eta1 = 0:0.01:xi;
modeshape1 = R(Om * eta1) * p2 + T(Om * eta1) * p3;
eta = xi + 0.11:0.01:1;
modeshape2 = R(Om * eta) * p2 + T(Om * eta) * p3 + T(Om * (eta - xi));
modeshape = [modeshape1, modeshape2];
modeshape = modeshape/max(abs(modeshape));
etat = [eta1, eta];
```

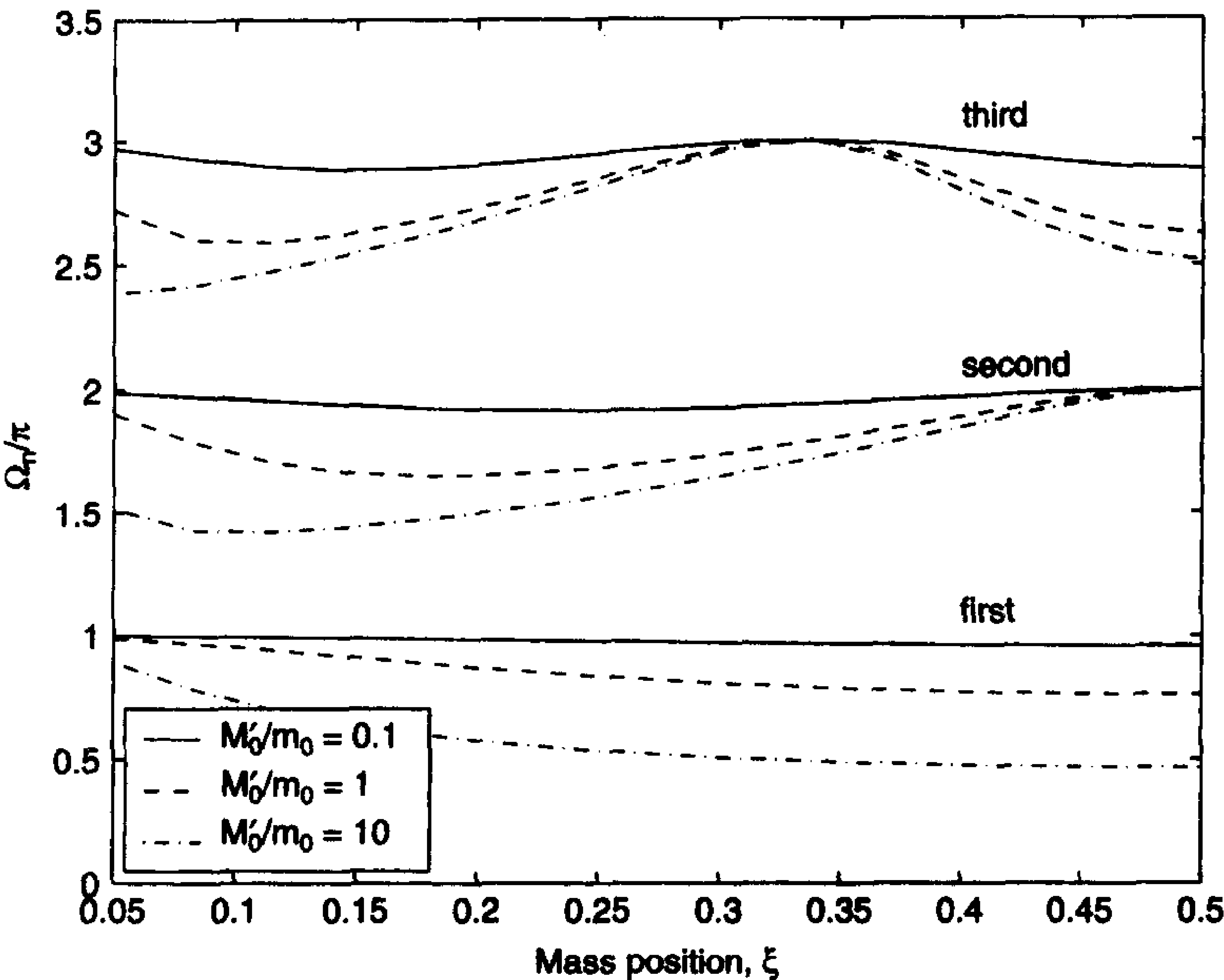


图 9.32 两端简支梁的 3 个最低固有频率系数随附加质量值及附加位置而变化的曲线

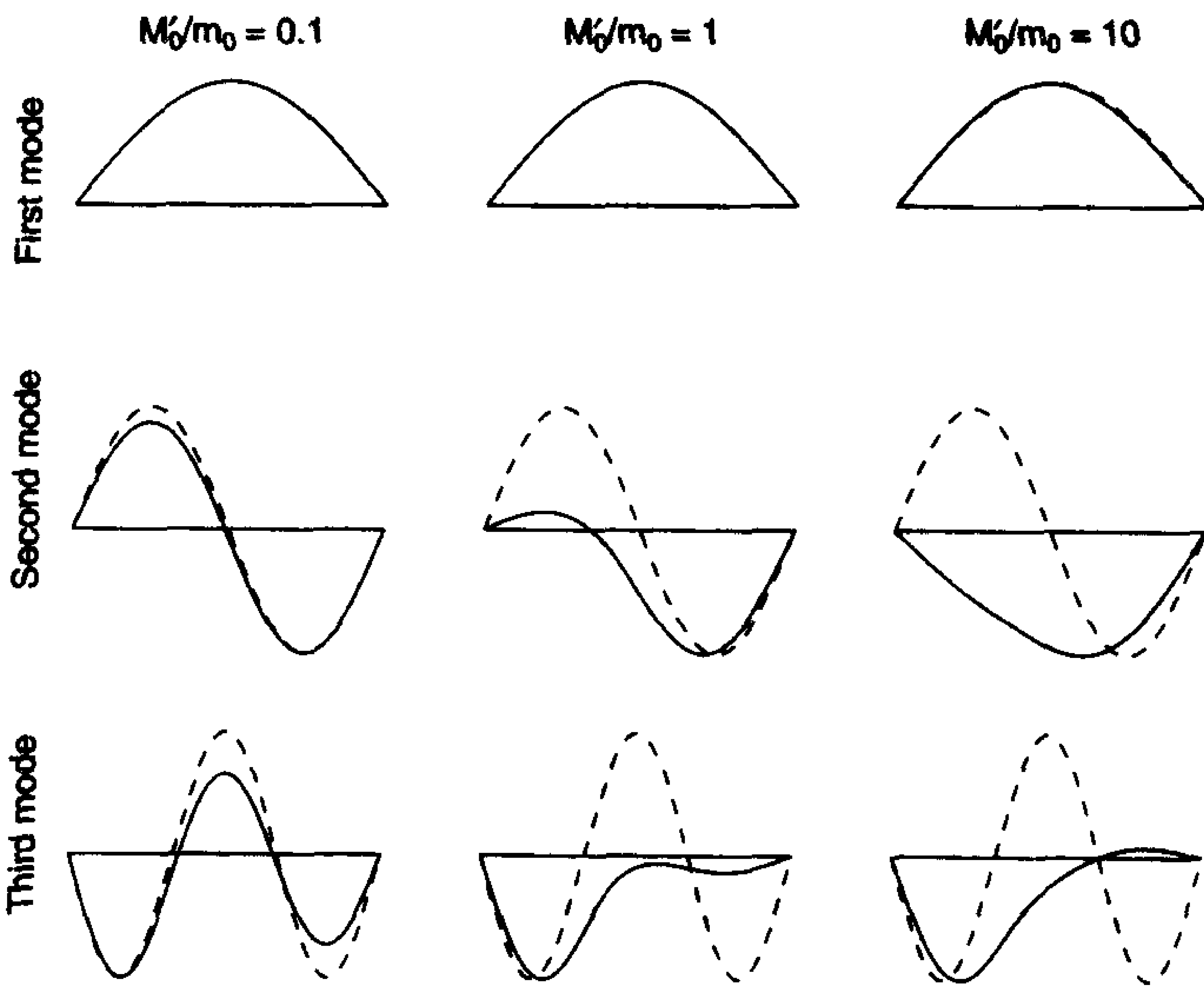


图 9.33 两端简支梁在 $\xi = 0.4$ 处附加质量得到的模态：实线表示带附加质量的梁；虚线表示不带附加质量的梁

9.5 应用函数一览

表 9.5 列出了本章介绍的函数及其说明。

表 9.5　第 9 章介绍的 MATLAB 函数

MATLAB 函数	说明
besseli	修改过的一类贝塞尔函数
besselj	一类贝塞尔函数
besselk	修改过的二类贝塞尔函数
bessely	二类贝塞尔函数
bode	线性非时变模型频率响应的波特图
damp	线性非时变模型的阻尼因子和自然频率
fminimax	求极小极大值
fminsearch	求无约束多变量函数的极小值
impulse	线性非时变模型的脉冲响应
rank	求完全矩阵的线性独立行数或列数(秩)
step	线性时不变模型的阶跃响应
tf	定义线性时不变模型的传递函数
trace	对矩阵对角线元素求和

练习

9.1 由式(9.1)给出的两质量系统,在给定初始条件 $r(0)=2.0, \mathrm{d}r(0)/\mathrm{d}\tau=0.0, \theta(0)=0.0$ 及 $\mathrm{d}\theta(0)/\mathrm{d}\tau=0.5$ 时,用数值方法求其在(r,θ)平面上的轨迹。用 ode45 函数求解时分别用到下列参数值:(a)每种状态下扫描时间取 20 单位,迭代步长为20/400,相对误差为 10^{-3},绝对误差为 10^{-3};(b)每种状态下扫描时间取 20 单位,迭代步长为 20/4000,相对误差为 10^{-6},绝对误差为 10^{-6}。

判断每种情况下在 20 个扫描单位内单位质量的角动量 $r^2\dot{\theta}$是否保持不变,并画出下列图形:(i)轨迹图,(ii)单位质量角动量对时间的关系图。

9.2 令 $r(0)=r_o$, $\mathrm{d}\theta(0)/\mathrm{d}\tau=\theta_o$。式(9.1)简化为一个方程式:

$$\frac{1}{2}\left(\frac{\mathrm{d}r}{\mathrm{d}\tau}\right)^2+\frac{r_o^2\theta_o^2}{2r^2}-\frac{4\pi^2}{r}=\frac{\theta_o^2}{2}-\frac{4\pi^2}{r_o}$$

令 $\mathrm{d}r/\mathrm{d}\tau=0$,由上述方程可求得半径 r 的最大值和最小值。判断一颗绕地球转动的卫星在下列初始条件下是否会坠落到地球表面:

a. 初始条件 1:

$$r(0)=2.0\quad \frac{\mathrm{d}r(0)}{\mathrm{d}\tau}=0\quad \theta(0)=0.0\quad \frac{\mathrm{d}\theta(0)}{\mathrm{d}\tau}=2.0$$

b. 初始条件 2:

$$r(0)=2.0 \quad \frac{dr(0)}{d\tau}=0 \quad \theta(0)=0.0 \quad \frac{d\theta(0)}{d\tau}=0.2$$

9.3 对式(9.8)的弹簧－质量－阻尼器系统的三种响应，用函数 **freeosc** 以数值方法求出欠阻尼系统、($\zeta=0.1$)临界阻尼系统($\zeta=1$)及过阻尼系统($\zeta=2.0$)的响应，所有情况下令 $x_0=1.0$, $v_0=1.0$ 并将求得的响应结果与下列方程计算出的结果进行比较[①]:

欠阻尼系统:

$$x(\tau)=e^{-\zeta\tau}\{x_o\cos(\sqrt{1-\zeta^2}\tau)+\frac{v_o+\zeta x_o}{\sqrt{1-\zeta^2}}\sin(\sqrt{1-\zeta^2}\tau)\}$$

临界阻尼系统:

$$x(\tau)=[x_o+(v_o+x_o)\tau]e^{-\tau}$$

过阻尼系统:

$$x(\tau)=C_1e^{(-\zeta+\sqrt{\zeta^2-1})\tau}+C_2e^{(-\zeta-\sqrt{\zeta^2-1})\tau}$$

其中，$\tau=\omega_n t$，而且:

$$C_1=\frac{x_o(\zeta+\sqrt{\zeta^2-1})+v_o}{2\sqrt{\zeta^2-1}}$$

$$C_2=\frac{-x_o(\zeta-\sqrt{\zeta^2-1})-v_o}{2\sqrt{\zeta^2-1}}$$

9.4 如表 9.6 所示的自由振动数据，假定响应具有以形式:

$$x(t)=\frac{X_o e^{-\zeta\omega_n}}{\sqrt{1-\zeta^2}}\sin(\omega_n t\sqrt{1-\zeta^2}+\varphi)$$

其中，

$$\varphi=\arctan\frac{\sqrt{1-\zeta^2}}{\zeta}$$

对这些数据进行曲线拟和，并求 X_o，ζ 和 ω_n。

表 9.6 自由振动数据

时间	幅度	时间	幅度	时间	幅度
0.000	0.801	10.00	-0.0151	20.77	0.0379
0.692	0.365	10.77	-0.006 88	21.54	0.0167
1.538	-0.386	11.54	0.118	22.31	-0.0184
2.308	-0.562	12.31	0.0882	23.08	-0.0259
3.077	-0.114	13.07	-0.028	23.85	-0.0141
3.846	0.349	13.85	-0.0871	24.61	0.0149
4.615	0.338	14.15	-0.0551	25.38	0.0115

① 参见 D.J Inman，出处同上，1.3 节；S.S.Rao，出处同上，2.6 节；B.Balachandran 和 E.B.Magrab，出处同上，4.2 节。

（续表）

时间	幅度	时间	幅度	时间	幅度
5.385	-0.301	15.85	0.0220	26.15	0.003 67
6.154	-0.204	16.23	0.0687	26.92	-0.0148
6.923	0.104	17.92	0.0376	27.69	-0.0125
7.692	0.228	18.46	-0.040	28.46	0.0157
8.461	0.008	19.23	-0.0514	29.23	0.002 63
9.231	-0.010	20.00	-0.006 41	30.00	-0.007 27

9.5　一质量为 m 的物体沿长为 l 的粗糙杆滑动，杆绕 O 点旋转。杆在水平面的转动角度用 $\theta(t)$ 表示，物体沿杆滑动，以旋转点开始计算的滑动位置用 r 表示。当杆在水平面以固定角速度 $\mathrm{d}\theta/\mathrm{d}t=\omega$ 旋转时，m 的振动方程为：

$$\ddot{r}+2\mu\omega\dot{r}-\omega^2 r=0$$

其中，μ 为杆与物体间的摩擦系数，$\mu=0.2$，$l=3.0$ m，$\omega=6$ rad/s，初始条件为：$r(0)=1.0$ m，$\theta(0)=0.0$ rad/s，$\dot{r}(0)=0.0$ m/s，在图中画出物体离开杆前在 (r,θ) 平面内经过的路径。

9.6　加速计的测量值与真实幅值之比为：

$$\frac{A_m}{A_t}=\frac{1}{\sqrt{(1-r^2)^2+4\zeta^2 r^2}}$$

其中，$r=\omega/\omega_n$，ω 为加速频率。ω_n 为加速计的固有频率，ζ 为加速计的阻尼因子。求：

(a) 画出 A_m/A_t 对 r 和 ζ 的关系图。

(b) 加速计的质量 $m=0.01$ kg，自然频率为 150 Hz，测量 6000 r/m 加速度时误差 e 在下列值的 ±2.0% 范围内①，求加速计的阻尼因子：

$$e=100\left(1-\frac{A_m}{A_t}\right)\%$$

9.7　一地震计测量值与真实幅值之比为②：

$$\frac{\mathrm{d}_m}{\mathrm{d}_t}=\frac{r^2}{\sqrt{(1-r^2)^2+4\zeta^2 r^2}}$$

其中 $r=\omega/\omega_n$，ω 为加速频率；ω_n 为地震计的自然频率。ζ 为地震计的阻尼因子：

(a) 求 d_m/d_t 随 r 和 ζ 变化的曲线图。

(b) 求地震计在 500 r/m 到 1500 r/m 测量范围内振动时，误差不超过 ±2.0% 的最大固有频率。

9.8　如图 9.34 所示的单自由度系统，在其平衡位置中心有一宽为 $2b$ 的死区。系统控制方程为：

$$\begin{aligned} m\ddot{x}+k(x+b)+2c\dot{x}&=F(t) && x<-b\\ m\ddot{x}+2c\dot{x}&=F(t) && -b\leqslant x\leqslant b\\ m\ddot{x}+k(x-b)+2c\dot{x}&=F(t) && x\geqslant b \end{aligned}$$

---

① 参见 D.J. Inman，出处同上，2.6 节；S.S. Rao，出处同上，10.3 节；B. Balachandran 和 E.B. Magrab，出处同上。

② 参见 S.S. Rao，出处同上，10.3 节；B. Balachandran 和 E.B. Magrab，出处同上。

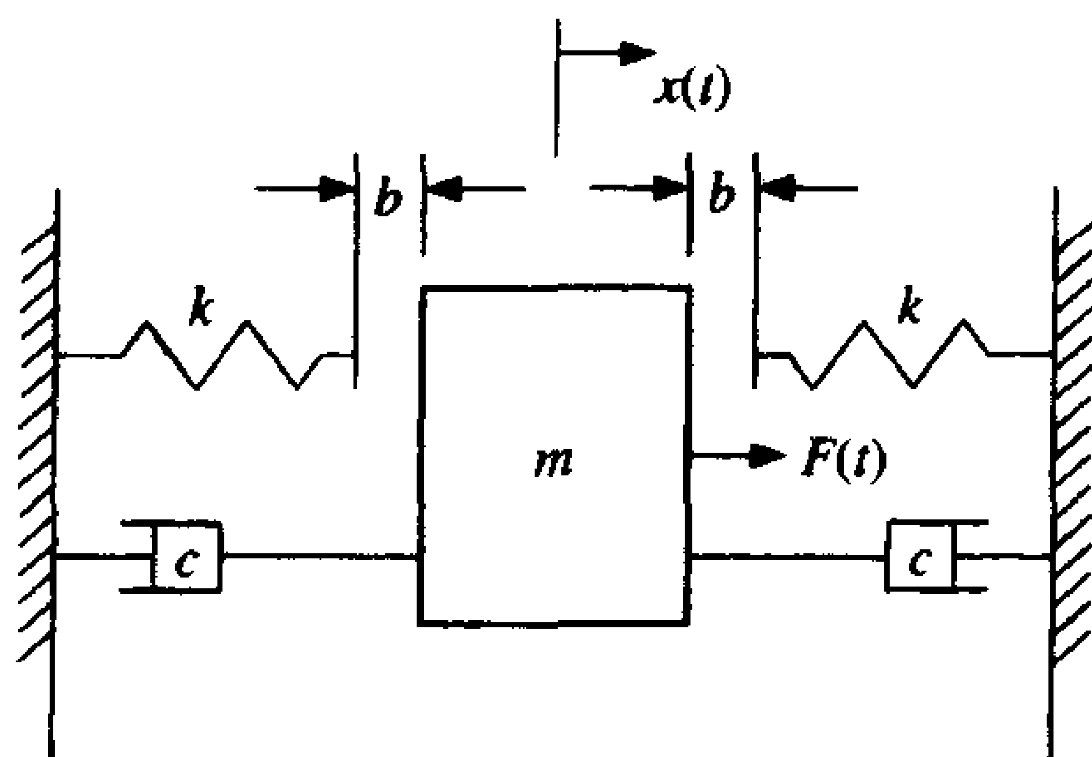


图 9.34 有死区的弹簧 - 质量 - 阻尼器系统

当 $m=10.0$ kg, $k=150\times10^3$ N/m, $c=50$ Ns/m, 初始条件为 $x(0)=0$, $\mathrm{d}x(0)/\mathrm{d}t=20$ m/s时, 求下列情况的自然响应: (a) 死区 $b=1.0$ μm, (b) $b=1000$ μm, 同时求出 $b=5.0$ μm, $x(0)=0$ m, $\mathrm{d}x(0)/\mathrm{d}t=0$ m/s 时系统在 $F(t)=20\cos(12t)u(t)$ N 下的强迫响应, 其中 $u(t)$为单位阶跃函数。

9.9 如图 9.35 所示的机械系统, 在底部受到激励 $x_b(t)$时的传递函数为:

$$G(s)=\frac{x_m(s)}{x_b(s)}=\frac{2\zeta\omega_n s+\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

令 $\zeta=0.1$, $\omega_n=4$ rad/s。用 bode 函数求出系统的振幅响应与相位响应, 将所求结果与用下式分析的结果进行比较:

$$G(\omega)=\sqrt{\frac{\omega_n^4+(2\zeta\omega\omega_n)^2}{(\omega_n^2-\omega^2)^2+(2\zeta\omega\omega_n)^2}}$$

$$\phi(\omega)=\tan^{-1}\left(\frac{2\zeta\omega}{\omega_n}\right)-\tan^{-1}\left(\frac{2\zeta\omega\omega_n}{\omega_n^2-\omega^2}\right)$$

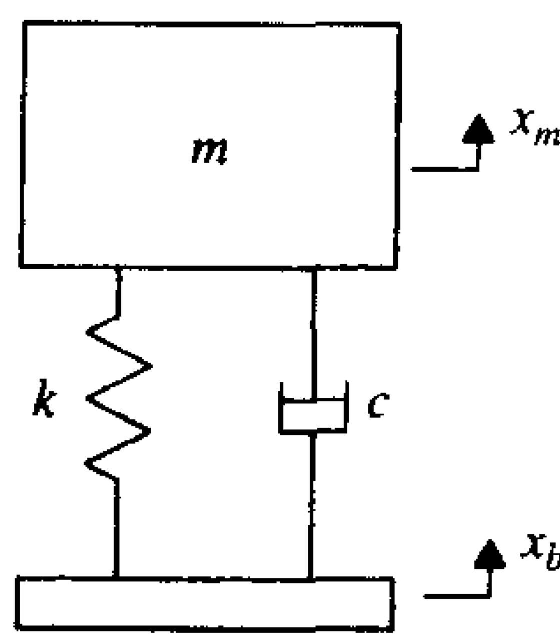


图 9.35 基部受激励的弹簧 - 质量 - 阻尼器系统

9.10 分析 9.3.2 节讨论的减振器, 在无阻尼的情况下, 即 $c_1=c_2=0$, 令第二级质量与主质量比值为 0.2, 第二级质量的自然频率与主质量的自然频率比值为 1.0, 画出 m_1 和 m_2 的无量纲频率响应关系图, 并验证: (a) 两图中谐振(极点)位置一致, (b) m_1 的响应曲线在减振器的自然频率处有一抗振点(零点)。

9.11 分析 9.3.2 节讨论的减振器, 令 $c_1=0$, 第二级质量与主质量之比固定为 0.05。主质量等于 1.0 单位, 主系统的自然频率为 1 rad/s, 减振器的自然频率与主质量的自然频率之比为 1.0。减振器阻尼因子取不同值时, 画出主质量的频率响应曲线图,

并验证不同图形在两个频率处相交①。

9.12 如图 9.36 所示的架空吊车,其小车通过钢索运输一质量为 m 的重物。当小车以加速度 $b(t)$移动时,吊车重物振动的控制方程为:

$$L\frac{d^2\theta}{dt^2}+g\sin\theta=-b(t)\cos\theta$$

其中,$g=9.8\ \mathrm{m/s^2}$ 为重力加速度。如果钢索长度为 2 m,在时间段 $0\leqslant t\leqslant 10$ s 内画出下列情况下的摇摆振动图 $\theta(t)$:

(a) $b(t)=10u(t)\ \mathrm{m/s^2}$, $\theta(0)=0.2$ rad 及 $d\theta(0)/dt=0$ rad/s。

(b) $b(t)=0.2u(t)\ \mathrm{m/s^2}$, $\theta(0)=0.2$ rad 及 $d\theta(0)/dt=0$ rad/s。其中,$u(t)$为单位阶跃函数。

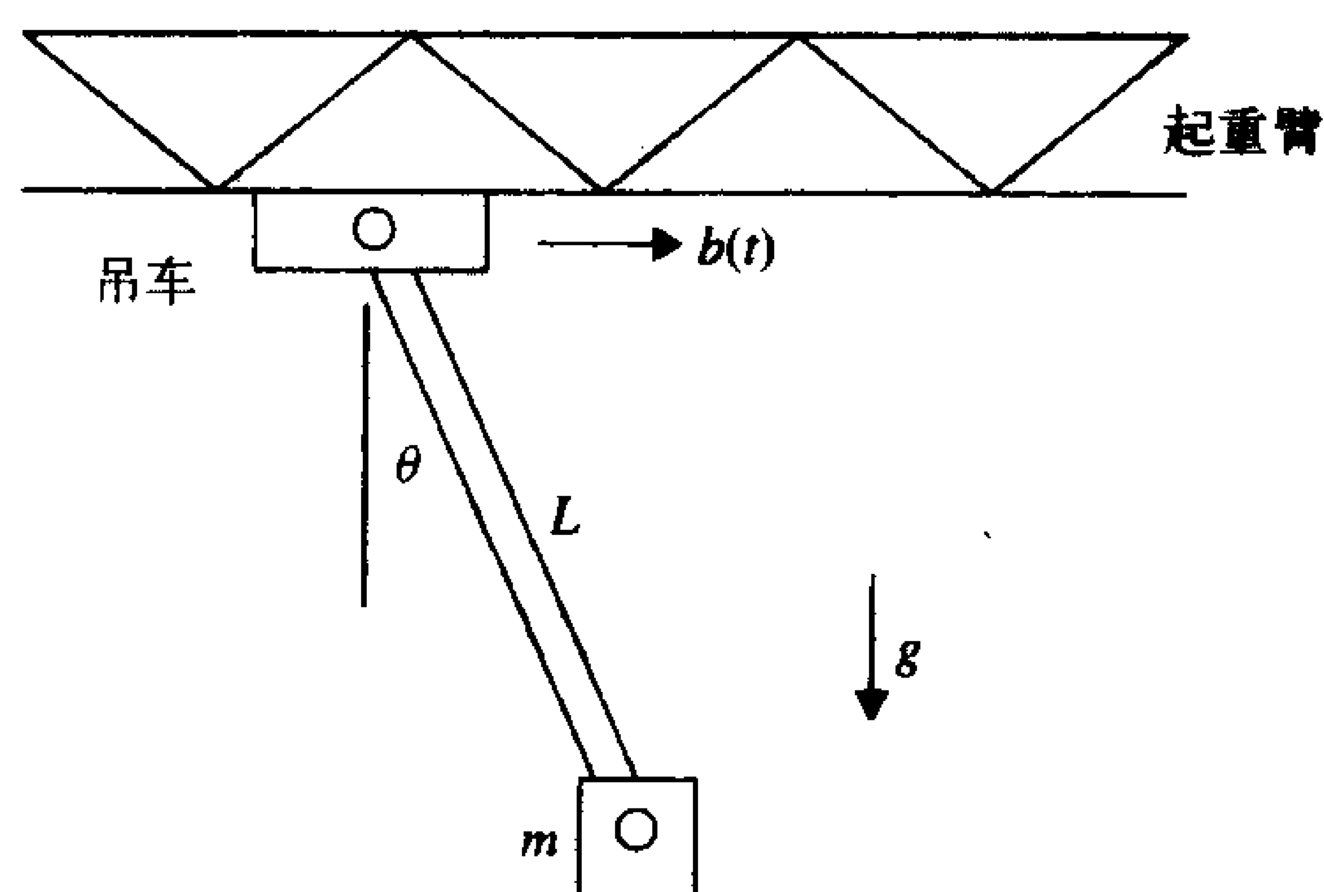


图 9.36　架空吊车上的小车运载一摇摆负载 m

9.13 图 9.37 所示的模型常用于分析车辆的弹性振动情况。该无阻尼系统的自由振动方程为:

$$m\frac{d^2x}{dt^2}+(k_1+k_2)x+(L_2k_2-L_1k_1)\theta=0$$

$$I_c\frac{d^2\theta}{dt^2}+(k_1L_1^2+k_2L_2^2)\theta+(L_2k_2-L_1k_1)x=0$$

如果 $k_1=1000$ lb/ft, $k_2=1500$ lb/ft, $L_1=5$ ft, $L_2=4$ ft, $m=50$ slug, $I_c=1000\ \mathrm{slug\cdot ft^2}$,用特征值问题求出自然频率、模态及中心点位置。

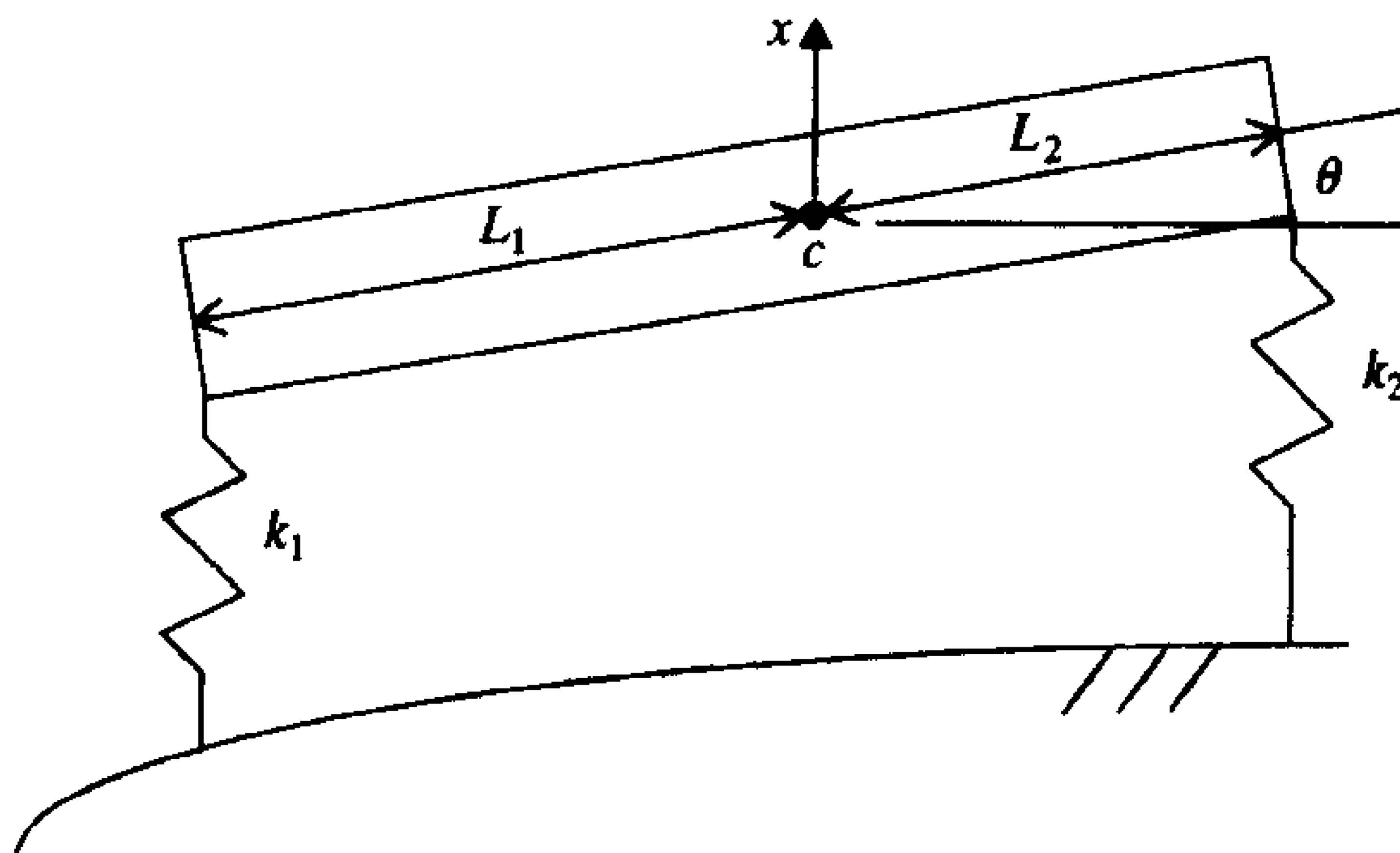


图 9.37　车辆的两自由度模型

① 参见 S.S.Rao,出处同上,9.10 节。

9.14 车辆悬挂系统可用图 9.38 所示的模型来表示。该系统的控制方程为:

$$m_1\frac{d^2x_1}{dt^2}+c_1\left(\frac{dx_1}{dt}-\frac{dx_2}{dt}\right)+k_1(x_1-x_2)=0$$

$$m_2\frac{d^2x_2}{dt^2}+c_1\left(\frac{dx_2}{dt}-\frac{dx_1}{dt}\right)+k_1(x_2-x_1)+k_2x_2+c_2x_2=k_2y+c_2\frac{dy}{dt}$$

在初始条件 $x_1(0)=0.5$ m, $dx_1(0)/dt=0$ m/s, $x_2(0)=0.2$ m, $dx_2(0)/dt=0$ m/s 时,令 $m_1=1$ kg, $m_2=2$ kg, $c_1=1$ N/m/s, $c_2=5$ N/m/s, $k_1=10$ N/m, $k_2=30$ N/m,求系统的固有响应。对振动方程做拉普拉斯变换,并假定初始条件为 0,画出系统的频率响应曲线图。

9.15 如图 9.39 所示,一循环转动刚圆轴两端受弹簧支撑,转子绕轴转动的角速度为 ω rad/s,转子沿转动轴方向有极惯性矩 J_p,转动平面内任何方向的横向惯性矩 J_t,各自水平方向的支撑刚性为 k_1 和 k_2。通过求解以下特征值问题可求得自由转动速度 Ω:

$$\boldsymbol{K}'w=\lambda\boldsymbol{M}'w$$

式中,$\lambda=\Omega^2$

$$\boldsymbol{K}'=\begin{bmatrix}KM^{-1}K & KM^{-1}G\\ G'M^{-1}K & K+G'M^{-1}G\end{bmatrix}\quad \boldsymbol{M}'=\begin{bmatrix}K & 0\\ 0 & M\end{bmatrix}$$

及

$$\boldsymbol{M}=\begin{bmatrix}m&0&0&0\\0&J_t&0&0\\0&0&m&0\\0&0&0&J_t\end{bmatrix}\quad \boldsymbol{G}=\begin{bmatrix}0&0&0&0\\0&0&0&-J_p\omega\\0&0&0&0\\0&J_p\omega&0&0\end{bmatrix}\quad \boldsymbol{K}=\begin{bmatrix}k_1&0&0&0\\0&k_2&0&0\\0&0&k_1&0\\0&0&0&k_2\end{bmatrix}$$

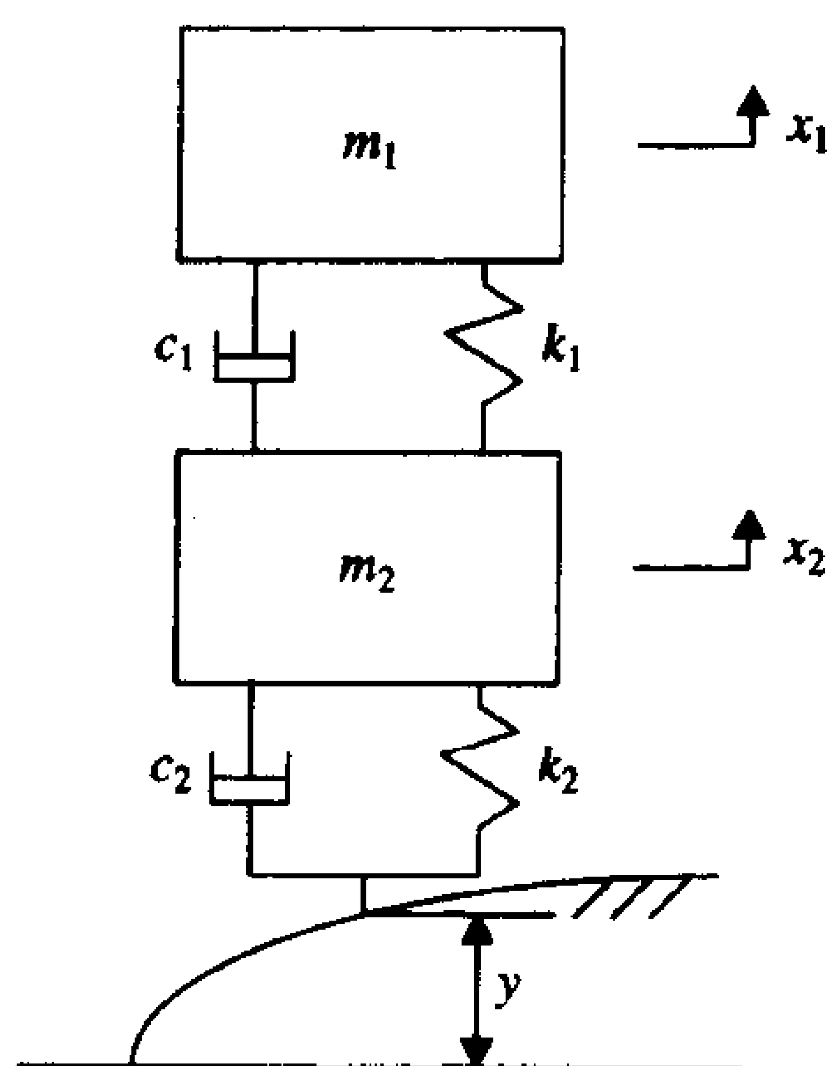


图 9.38 车辆悬挂系统的四分之一模型

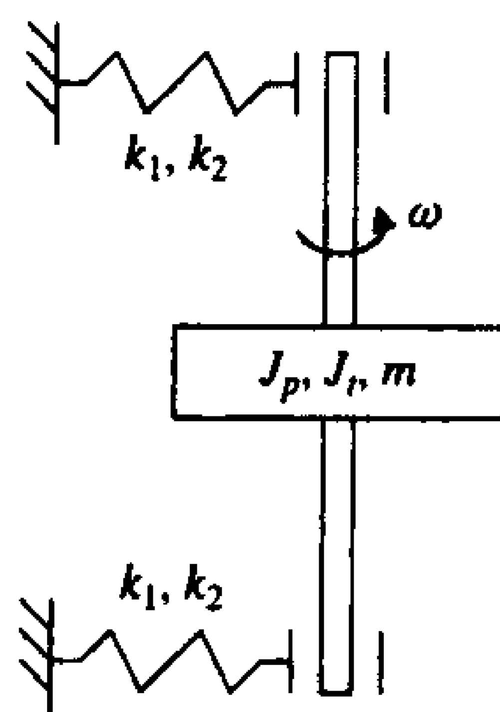


图 9.39 在弹性力支撑下的刚性转动转子

设 $m=10$ kg, $J_p=2$ kg·m², $J_t=1.2$ kg·m², $k_1=k_2=2.5\times10^6$ N/m,在图中画出随 Ω 变化的曲线,其中 $0\leqslant\omega\leqslant1500$ rad/s。该图为坎贝尔(Campbell)图表[①] 的一个例子。在 $\Omega=\omega$ 处的速度称为临界速度,结果应如图 9.40 所示。

① 参见 G. Genta, *Vibration of Structures and Machines: Practical Aspects*, Springer-Verlag, New York, NY, 1993 年, 4.3 节。

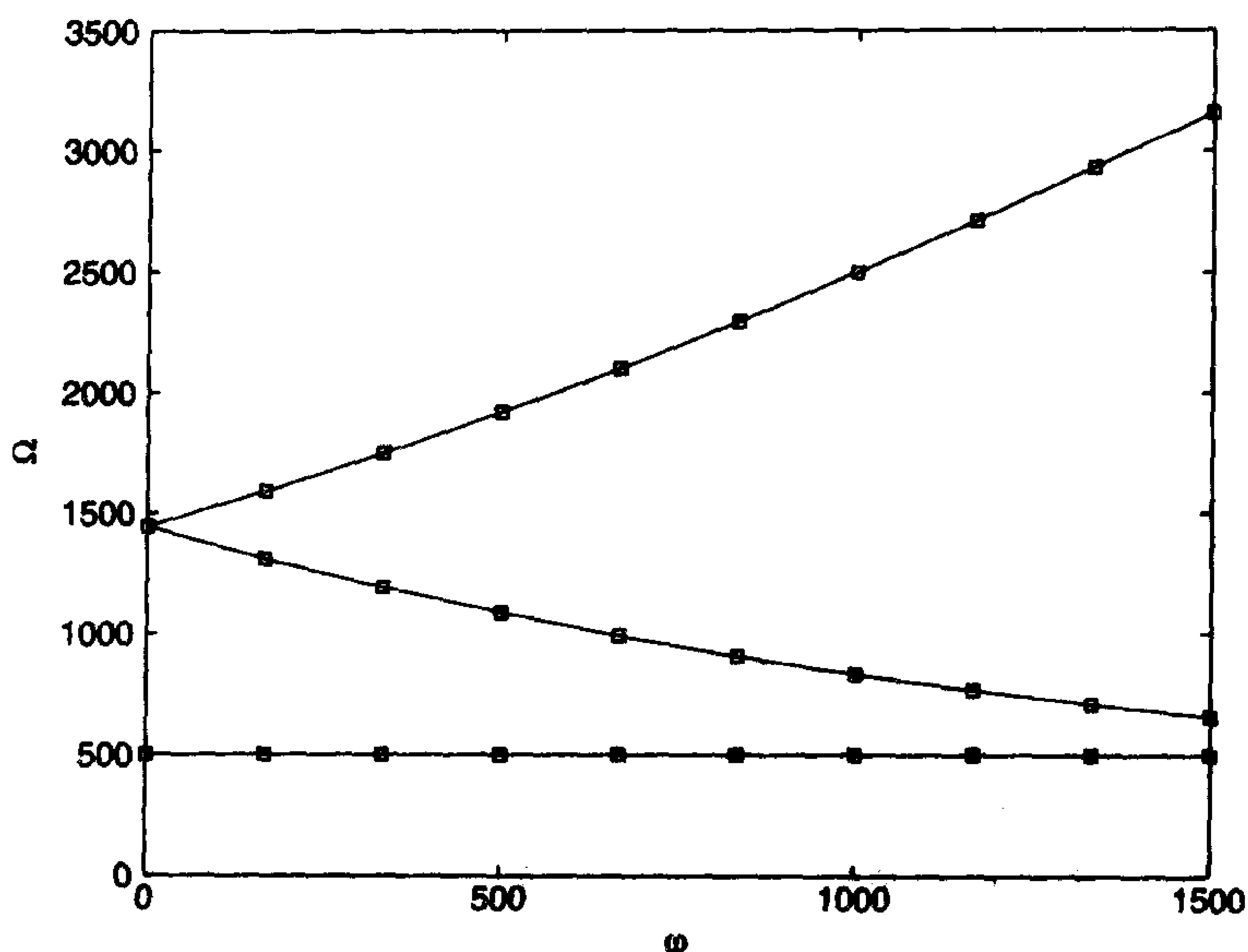


图9.40 在弹性力支撑下刚性转子的坎贝尔图表

参考文献

1. B. Balachandran and E. B. Magrab, *Vibrations*, Thomson Brooks/Cole, Belmont CA, 2003.
2. D. T. Greenwood, *Principles of Dynamics*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 1988.
3. F. J. Hale, *Introduction to Space Flight*, Prentice Hall, Englewood Cliffs, NJ, 1994.
4. D. J. Inman, *Engineering Vibration*, Prentice Hall, Englewood Cliffs, NJ, 1994.
5. E. B. Magrab, *Vibrations of Elastic Structural Members*, Sijthoff & Noordhoff, Alphen aah den Rijn, The Netherlands, 1979.
6. L. Meirovitch, *Elements of Vibration Analysis*, McGraw-Hill, New York, NY, 1986.
7. F. C. Moon, *Applied Dynamics with Applications to Multibody and Mechatronic Systems*, John Wiley & Sons, New York, NY, 1998.
8. A. H. Nayfeh and B. Balachandran, *Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods*, John Wiley & Sons, New York, NY, 1995.
9. S. S. Rao, *Mechanical Vibrations*, 3rd ed., Addison-Wesley, Reading, MA, 1995.
10. B. H. Tongue, *Principles of Vibration*, Oxford University Press, New York, NY, 1996.

第10章 控制系统

本章介绍使用 MATLAB 控制工具箱和 SIMULINK 对控制系统进行描述、设计和计算。

10.1 控制系统设计概述

研究图10.1所示的控制系统可以发现,典型控制系统中一般包括用微分方程表示的设备模型(控制对象),并具有一个或多个控制变量[输入 $u(t)$]和一个或多个输出 $y(t)$。对所有控制系统而言都有一个算法控制器,接收操作命令[参考输入 $r(t)$]和对象输出,并计算控制变量。图10.1描述了串联控制系统,在此控制系统中控制器取得参考输入和输出之间的差,形成误差信号 $e(t)=r(t)-y(t)$,根据误差信号再产生对象输入 $u(t)$。控制器一般由计算机实现,但是在一些传统系统中还采用模拟电路或机械装置实现。

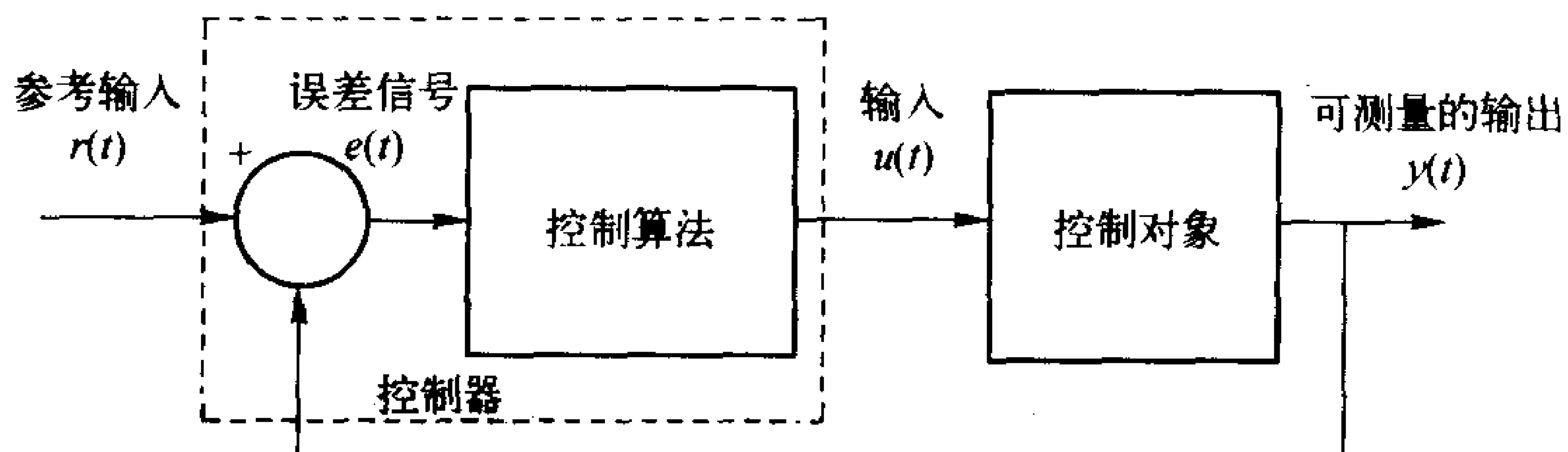


图10.1 反馈回路方框图

在控制系统中,首先应该明确系统的控制目标是什么。控制系统设计目标是改变物理系统使其符合要求。这不仅要求预知给定系统模型的变化,而且还要求能达到预期的目标。因此,控制系统设计存在着矛盾。

控制系统设计目标分为三个方面的内容,即性能、安全性和健壮性。例如对伺服电机系统而言,在性能方面,其对命令的响应度可量化为暂态设计目标,而对给定命令的最终逼近度可量化为稳态设计目标。在安全性方面,要求设计人员应确保系统工作在安全状态,避免进入危险工况或造成对自身的损伤。在健壮性方面,因为有时系统某些输入量不可测或不可控,或者当时间、过程或运行工况发生变化时被控系统进入不确定状态,因此应考虑系统设计的不确定性问题。如上述伺服电机系统,无论系统由风力驱动还是由太阳能驱动,都要求系统具备连续工作能力。因此,成功的控制系统设计对用户来说应该是不可见的,并拥有良好的性能、安全性和健壮性。

控制设计问题通常用反馈来说明。反馈的基本思想是用系统输出校正输入,使系统的运行不需人工干预。反馈系统除这个明显的优点外,还能减少非线性、增加系统的健壮性、提高系统的稳定性。反馈对控制工程师而言是非常有效(但不是惟一)的手段。其他有用的方法包括嵌入式系统程序设计、数字信号处理和系统辨识。

本章着重介绍如何使用 MATLAB 中的许多工具解决各种控制问题。10.2节详细介绍在 MATLAB 中如何使用传递函数方法、方框图法及状态空间描述方法描述控制系统。同时回顾

了这些模型在离散时间域的描述，亦即在嵌入式计算控制系统中将加以讨论。10.3节介绍控制系统的阶跃响应。10.4节详细介绍波特图、根轨迹和LQR/LQG等设计方法。最后，10.5节专门介绍设计实例。

10.1.1 控制器设计方法

控制器设计要求熟练掌握对描述实际系统特性的常微分方程的求解方法。然而在像MATLAB这样的计算工具出现以前，要想熟练掌握这一方法必须求解上百道微分方程问题。MATLAB中对线性时不变常微分方程有三种不同的描述方法：

1. 状态空间方程
2. 传递函数
3. 方框图

状态空间描述是在时域内用矩阵描述的，传递函数是在拉普拉斯域内使用复变量 s 的多项式描述的，在MATLAB中方框图描述是通过SIMULINK工具箱形象地描述输入和输出的连接。各种描述法之间的转换由MATLAB内置函数提供。

本章将讨论三种类型的控制目标：

1. 暂态
2. 稳态
3. 稳定性

暂态设计要求关注系统的短期行为，如敏感性和刚性。稳态设计要求关注系统的长期行为及一段时间内系统是如何运行的。标准输入信号有阶跃函数、斜坡函数和正弦函数，这些函数常用来检验系统是否满足暂态及稳态设计的要求。MATLAB中提供了可得到系统标准信号响应的函数。暂态和稳态要求与性能有关，而反馈稳定性更多是与安全有关。反馈对一个物理特性很好的系统而言，可能使系统稳定也可能使系统不稳定，而在实际中必须避免不稳定因素。暂态特性和稳态特性对设计要求通常是矛盾的，因此设计时应在两者之间进行折中。实际上，控制性能的限制来自传感器、执行机械和对象本身的限制。

用来解决控制问题的图解法包括：波特图、奈奎斯特(Nyquist)曲线和根轨迹，线性代数方法用于更先进的设计技术，如LQG方法、H_∞ 和 μ 综合方法等。对于大多数单入单出控制系统，使用一种控制器就能完成设计任务。

10.1.2 命名和文件约定

在本章中，使用了一套标准的命名约定。因为即使是一个简单的微分方程也需要多个向量和矩阵描述，所以MATLAB提供了一种方法，将必要的矩阵和向量集中到一个名称下。这种矩阵、向量甚至字符串的集合称为系统。*Plant* 用于标识系统名，在控制器设计过程中其结构是不变的，而 *Control* 用于标识将要设计的部分。最后的闭环系统由 *Plant* 及 *Control* 组成，用 *clSys* 标识。若断开反馈连接，则称系统为 *olSys*。本章中用来构成和分析控制系统的MATLAB函数将系统看做参数而不是看做向量和矩阵。

本章中研究几个系统实例，为方便起见，要创建用于返回系统模型的函数文件，函数返回

系统对象。实例包括：

- 负载的永磁电机(**MotorSS**)
- 弹性轴驱动的指针(**Pointer**)
- 磁悬浮器(**MagLev**)
- 倒置摆(**Pend**)
- 飞轮(**Fly**)

根据需要可设计出超前、滞后、比例积分(PI, proportional integral)和比例微分(PD, proportional derivative)等多种形式的控制器。

10.2 MATLAB 中系统的描述

控制系统的输入用实函数 $r(t)$描述，这个函数一般用于描述控制中的某些变量，如力、电压或温度。控制系统的输出用实函数 $y(t)$描述。函数值是一些可以测量的值，如角度、压力或速度。在控制系统中输入函数 $u(t)$和输出函数 $y(t)$之间的关系用线性时不变常微分方程表示，一般形式为：

$$a_n\frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n}+a_{n-1}\frac{\mathrm{d}^{n-1}y(t)}{\mathrm{d}t^{n-1}}+\ldots+a_0 y(t)=b_m\frac{\mathrm{d}^m u(t)}{\mathrm{d}t^m}+b_{m-1}\frac{\mathrm{d}^{m-1}u(t)}{\mathrm{d}t^{m-1}}+\cdots+b_0 u(t) \quad (10.1)$$

其中 $n \geqslant m$。方程的系数 a_j 和 b_j 是定常实数。MATLAB 中将系统模型存储成对象，类似于第 6 章和第 7 章的图形对象，通过使用 get 和 set 函数很容易得到这些模型的特性。用式(10.1)所示的微分方程形式表示典型的控制问题，MATLAB 提供了三种输入 - 输出关系的类。

- 传递函数描述(类 tf)
- 状态空间描述(类 ss)
- 零点 - 极点 - 增益描述(类 zpk)

因为控制回路经常由计算机实现，所以离散线性系统在实际中用得很多。上述三种描述都有相应的离散时间形式，采样时间是附加的信息。将系统模型封装成对象，允许用户将相关数据追加到描述中。与追加数据相关的例子包括 InputName，OutputName 和 Notes。

为了举例说明三种基本的描述，研究图 10.2 所示的直流永磁伺服电机的简单模型。输入电压 $v(t)$作用于电动机的线圈上，产生电流 $i(t)$。线圈上的电流在转子上产生转矩，且与电流大小成比例。有效惯量 J 是负载惯量 J_l 和转子惯量 J_m 的和。具有角度 $\theta(t)$的转子就像发电机一样，产生与转子角速度成比例的反向电压。根据力矩平衡原理及电路分析原理可以得到下面两个线性常微分方程①，用来描述输入电压 $v(t)$和输出角度 $\theta(t)$之间的关系：

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t}+k_b\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}+Ri(t)=v(t) \quad (10.2)$$

$$J\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2}+b\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}-k_\tau i(t)=0$$

① 参见 D.K.Anand 和 R.B.Zmood, *Introduction to Control Systems*, 第三版, Butterworth-Heinemann Ltd. Oxford, England, 1995 年。

与电有关的常数是 R, L, k_τ 和 k_b，其中 R 是电动机的阻抗，L 是线圈的电感，k_τ 是从电流到转矩的转换因数，而 k_b 是发电机的反电动势常数。总惯量 J 通常取决于负载惯量 J_l。若没有变速箱，电机摩擦系数 b 一般很小。下面将这些方程用三种不同形式的模型表示。

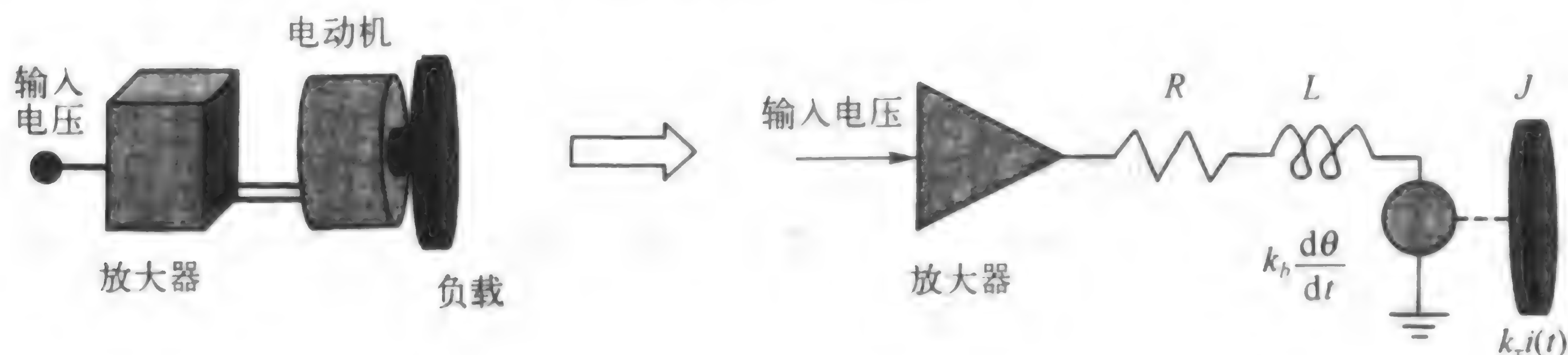


图 10.2　典型的伺服电机

10.2.1　状态空间模型

随着计算机的广泛应用，状态空间模型也得到了普遍应用，因为它比传递函数更具有数值稳定性。状态空间模型是相关的一阶微分方程组。因为存在 $\theta(t)$ 的二阶导数，所以式(10.2)所示的电动机模型不是一阶的。为了描述电机的状态空间模型，必须先将方程转化成 5.5.3 节那样的一阶形式。设：

$$x_1(t) = \theta(t)$$

$$x_2(t) = \frac{d\theta(t)}{dt} = \omega(t)$$

$$x_3(t) = i(t)$$

则式(10.2)变为：

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -\frac{b}{J}x_2 + \frac{k_\tau}{J}x_3$$

$$\frac{dx_3}{dt} = -\frac{k_b}{L}x_2 - \frac{R}{L}x_3 + \frac{v(t)}{L}$$

如果令：

$$x(t) = [x_1(t), x_2(t), x_3(t)]'$$

$$u(t) = v(t)$$

$$y(t) = \theta(t) = x_1(t)$$

则电机系统的状态空间描述为：

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \qquad (10.3)$$

$$y(t) = Cx(t) + Du(t)$$

其中，

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -b/J & k_\tau/J \\ 0 & -k_b/L & -R/L \end{bmatrix}$$

$$B = [0 \quad 0 \quad 1/L]'$$
$$C = [1 \quad 0 \quad 0]$$
$$D = [0]$$

在 MATLAB 中,矩阵 A, B, C, D 是描述微分方程的必要数据。

下面通过几个例子说明这些结果。

例 10.1 伺服电机的状态空间模型

为方便地将模型输入到 MATLAB 中,可创建函数 **MotorSS** 来返回图 10.2 所示伺服电机的状态空间系统模型。假设参数 $L = 5$ mH(电机电感),$R = 5\Omega$(电机电阻),$k_b = 0.125$ V/rad/s(反电动势),$k_\tau = 15$ Nm/A(电机转矩常数),$J = 0.03$ kg·m^2(电机惯性)和 $B = 0.01$ Nm/rad/s(电机摩擦系数),函数如下:

```
function Plant = MotorSS(J1)
if nargin < 1
  J1 = 0;
end;
L = 5e - 3;R = 5;kb = 12.5e - 2;
ki = 15;J = 3e - 2 + J1;b = 1e - 2;
A = [0 1 0;0, - b/J,ki/J;0, - kb/L, - R/L];
B = [0;0;1/L];
C = [1,0,0];
D = 0;
Plant = ss(A,B,C,D);
set(Plant,'InputName','volts',OutputName',' \ theta');
set(Plant,'StateName',{' \ theta',' \ omega','i'});
set(Plant,'Notes','Small DC servomotor');
```

函数 ss 将矩阵 A, B, C 和 D 代入到单个系统对象中。键入

```
MotorSS
```

在 MATLAB 命令窗口中显示:

```
a =
              \theta    \omega      i
     \theta     0         1         0
     \omega     0      -0.33333    500
        i       0        -25      -1000
b =
              volts
     \theta     0
     \omega     0
        i      200
c =
              \theta    \omega      i
     \theta     1         0         0
d =
              volts
     \theta     0
Continuous-time model.
```

在默认情况下，输入用 $u1$ 表示，输出用 $y1$ 表示，内部状态变量为 $x1$，$x2$ 和 $x3$。可以用 set 函数提供这些标号，并将状态空间变量命名为 θ，ω 和 i。此外，还要有注释说明模型所描述的内容。若没有对模型进行标志，则在 MATLAB 命令窗口中显示以下内容：

```
a=
            x1      x2          x3
      x1    0       1           0
      x2    0    -0.33333      500
      x3    0      -25       -1000
b=
            u1
      x1    0
      x2    0
      x3   200
c=
            x1      x2          x3
      y1    1       0           0
            u1
d=
      y1    0
Continuous-time system.
```

当函数 Motorss 调用时，只返回系统对象，而不返回常数。如果想重新得到矩阵 A、B、C 和 D，则可以使用：

```
[A,B,C,D] = ssdata(MotorSS)
```

在命令窗口显示 A，B，C 和 D：

```
A=
  1.0e+003*
          0    0.0010       0
          0   -0.0003    0.5000
          0   -0.0250   -1.0000
B=
   0
   0
  200
C=
   1  0  0
D=
   0
```

MATLAB 提供的内置函数可以将 MotorSS 这样的系统对象看做参数。将电机接成如图 10.3 所示的简单反馈系统中，如果要验证电机的特性，由框图可得：

$$v(t) = r(t) - y(t) = r(t) - \boldsymbol{C}x(t)$$

则式(10.3)相应地变成：

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = (\boldsymbol{A} - \boldsymbol{BC})x(t) + \boldsymbol{B}r(t)$$

$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}r(t)$$

就是将式(10.3)中的 $\boldsymbol{A}$ 用 $\boldsymbol{A} - \boldsymbol{BC}$ 代替。MATLAB 提供的数学运算函数命令如下：

```
clSys = feedback(MotorSS,1);
```

将返回闭环系统。第二个参数中的数值 1 表示反馈回路的传递函数为 1。注意 *clSys* 中的 *A* 等于 **MotorSS** 中的 ***A*** - ***BC***。可以通过输入下面的程序来检验：

```
clSys = feedback(MotorSS,1);
Plant = MotorSS;
clSys.a - (Plant.a - Plant.b * Plant.c)
```

返回(3×3)阶的零矩阵。

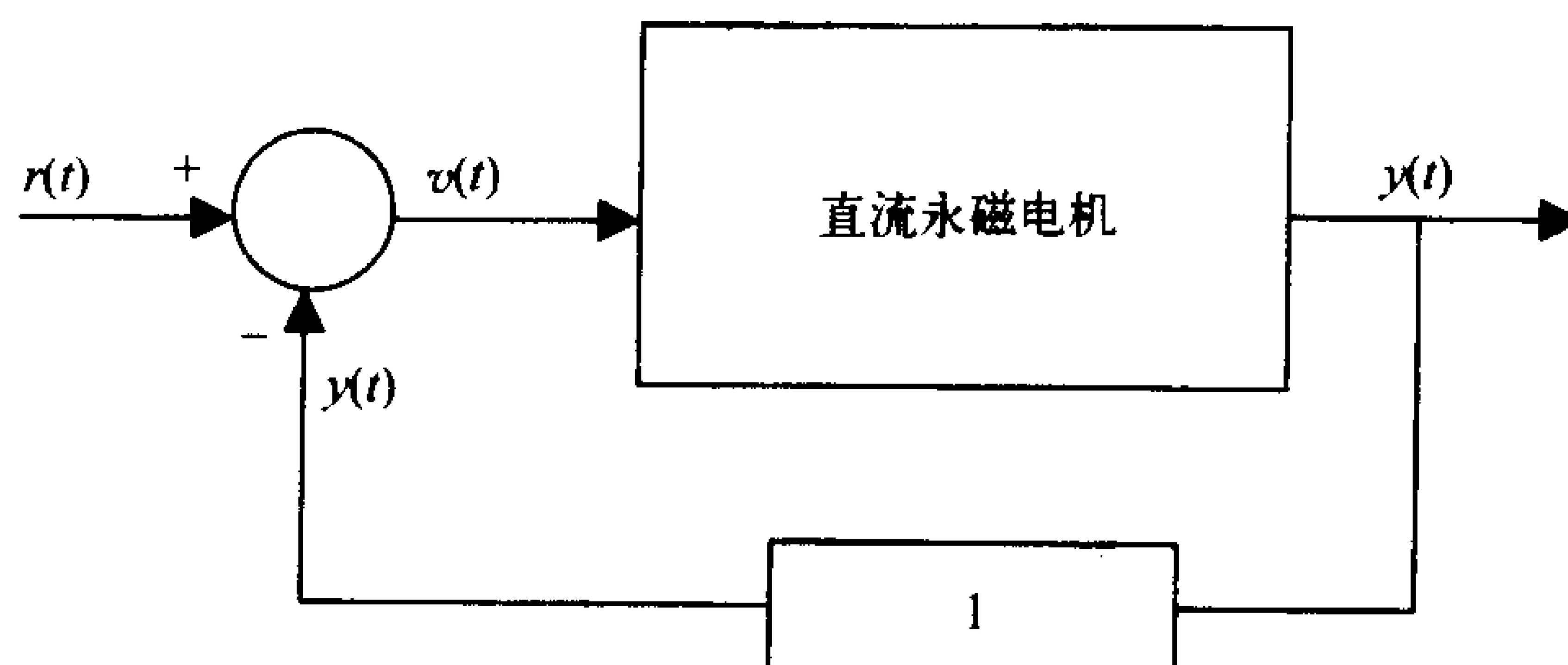


图 10.3　可控伺服电机的单位反馈控制系统

MATLAB 的函数 feedback 将算法与电机系统连成一个新的结构。在有输入时，将系统视为参数的 MATLAB 其他函数将用于求解微分方程。在下例中，使用 step 函数计算阶跃输入信号的系统响应。

例 10.2　伺服电机的阶跃响应

求图 10.2 所示伺服电机的阶跃响应程序为：

```
[y,t] = step(feedback(MotorSS,1));
plot(t,y,'k-');
xlabel('Time(s)');
ylabel('Rotor angle \ thet(t))(radians)');
```

执行后，曲线如图 10.4 所示。

扩展前面的程序产生负载惯量变化时的伺服电机阶跃响应。程序为：

```
t = 0:0.05:2;
Jload = 0:0.01:0.1;
data = zeros(length(t),length(Jload));
for i = 1:length(Jload)
   data(:,i) = step(feedback(MotorSS(Jload(i)),1),t);
end;
mesh(Jload,t,data);
view([45,30]);
xlabel('Load inertia J_1(kg m^2)');
ylabel('Time(s)');
zlabel('Rotor angle \ theta(t)(radians)')
```

曲线如图 10.5 所示。随着负载惯量 J_{load} 的增长，响应曲线出现了超调。在 10.3 节中，根据函数零极点的位置可得出系统响应的期望值。

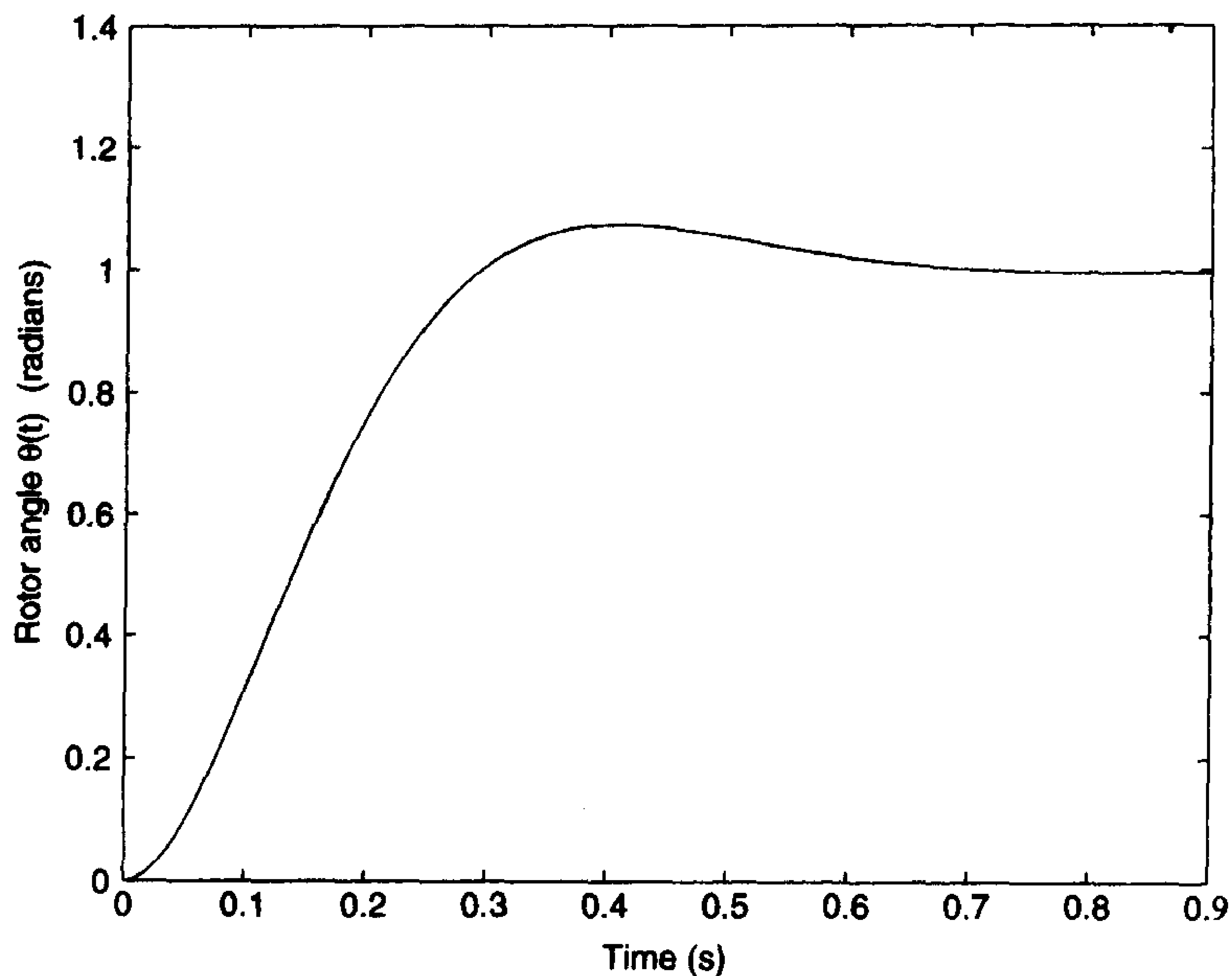


图 10.4　伺服电机控制系统的阶跃响应

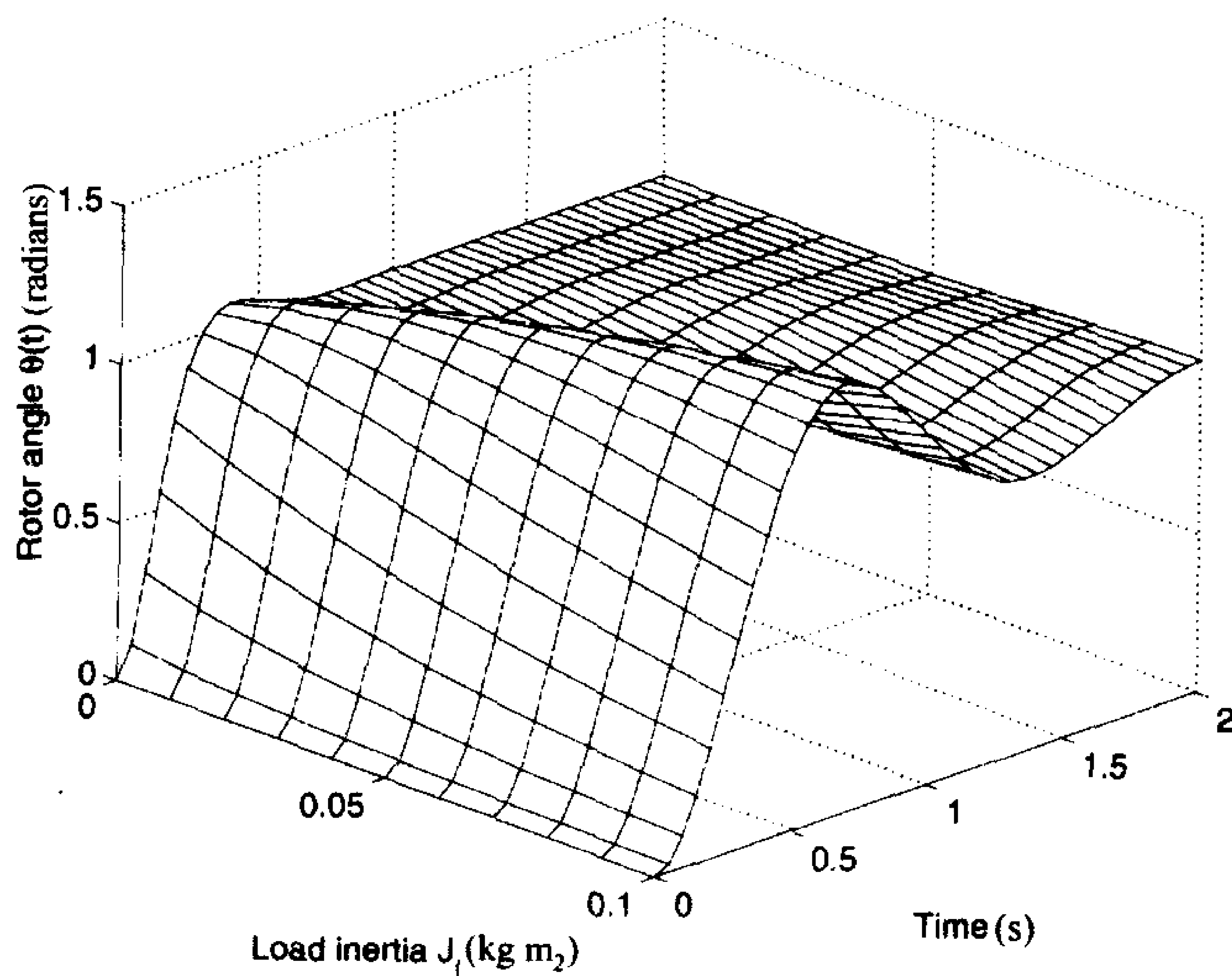


图 10.5　在不同负载惯量条件下闭环伺服电机的阶跃响应

10.2.2　传递函数描述

传递函数描述是指在零初始条件下，输出的拉氏变换除以输入的拉氏变换的结果。尽管这种描述不如状态空间方法应用普遍，而且对数值误差敏感，但是对设计者而言仍然是非常直观的[①]。式(10.1)所示的常微分方程的传递函数由下式给出：

---

① 参见 *MATLAB Control System Toolbox, User's Guide*，第四修订版，The Math Works, Natick, MA, 1992 年。

$$H(s)=\frac{Y(s)}{R(s)}=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_0}{a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0} \tag{10.4}$$

其中，$a_n \neq 0$ 且 $n \geq m$，分母多项式的根称为系统的极点，分子多项式的根称为系统的零点。在 MATLAB 中用两个向量描述传递函数，每一个向量分别包含 s 的系数。分子多项式系数为：

$$\text{Num}=[\text{bm},\cdots,\text{b1},\text{b0}]$$

分母多项式系数为：

$$\text{Den}=[\text{an},\cdots,\text{a1},\text{a0}]$$

可以用 a_n 除以每个系数使 a_n 为 1，但这不是必须的。为了得到传递函数 $H(s)$，应对方程进行拉氏变换及必要的代数运算。对式(10.2)进行拉氏变换得到：

$$k_b s\Theta(s)+(sL-R)I(s)=V(s)$$

$$(Js^2+bs)\Theta(s)-k_\tau I(s)=0$$

假设初始条件为零，求解 $\Theta(s)$ 得：

$$\Theta(s)=\frac{k_\tau}{(sL+R)(Js^2+bs)+k_\tau k_b s}\quad V(s) \tag{10.5}$$

系统传递函数模型可以由下面的程序产生，将 **MotorSS** 产生的状态空间模型转换为传递函数模型：

```
function PlantTF = MotorTF
L = 5e - 3; R = 5; kb = 12.5e - 2;
ki = 15; J = 3e - 2; b = 1e - 2;
Num = ki;
Den = conv([L,R],[J,b,0]) + [0,0,kb * ki,0];
PlantTF = tf(Num,Den);
```

键入下面的命令：

```
MotorTF
```

在 MATLAB 命令窗口中将会显示：

```
Transfer function:
              15
    ---------------------------
    0.00015 s^3 + 0.15s^2 + 1.925 s
```

正如矩阵 ***A***，***B***，***C*** 和 ***D*** 是状态空间实现的必要数据一样，分子多项式的系数 *Num* 和分母多项式的系数 *Den* 是传递函数得以实现的必要数据。输入、输出和其他描述部分的名称可以通过使用 set 和 get 函数来设置。使用下面的命令可以从分子和分母多项式中获得系数：

```
[Num, Den] = tfdata(MotorTF,'v')
```

字符 v 说明函数将 *Num* 和 *Den* 以行向量形式返回，若省略 v，则分子将以元数组的形式返回。元数组主要用来描述多入多出(MIMO, multiple-input, multiple-output)系统。多数情况下，我们仅讨论单入单出(SISO, single-input, single-output)系统。执行后，可得

```
Num =
  0  0  0  15
```

```
Den =
  0.0001   0.1500   1.9250        0
```

与状态空间系统一样,MATLAB 函数也可以将传递函数作为输入参数。例如:命令

```
step(feedback(MotorTF,1))
```

将会计算并绘出控制电机的阶跃响应曲线。注意命令 step 和 feedback 与前面状态空间模型使用的命令相同,这些函数可在任何系统对象描述方法中运行。

另外两个有用的函数是 pole 和 tzero,函数 pole 返回极点(传递函数中分母多项式的根)。例如:

```
p = pole(feedback(MotorTF,1))
```

显示:

```
p =
1.0e+02*
-9.8744
-0.0645+0.0773i
-0.0645-0.0773i
```

这是一个简单闭环控制系统的极点。函数 tzero 返回对象的传递零点,即

```
z = tzero(feedback(MotorTF,1));
```

若 z 为空矩阵,则说明系统没有零点。函数 pole 和 tzero 也可以用于状态空间系统。通过下式可实现传递函数与状态空间模型之间的转换:

$$H(s) = C(sI - A)^{-1}B$$

其中 $\boldsymbol{I}$ 是单位矩阵。从方程可以看出,分母的根等价于行列式$(s\boldsymbol{I}-\boldsymbol{A})$的根,也是矩阵 $\boldsymbol{A}$ 的特征值。极点、零点和特征值在分析控制系统特性中是经常用到的。

传递函数的一种描述形式是零点 - 极点 - 增益,其参数是零点和极点。通常传递函数是一个有理多项式函数,所以其分子和分母表达式都可以分解成下面的形式:

$$H(s) = k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \tag{10.6}$$

传递函数由极点和零点及增益常数 k 惟一地确定。要生成如图 10.3 所示的闭环控制系统的零点 - 极点 - 增益模型,首先产生极点和零点,通过除以直流增益将增益设置为 1。程序为:

```
Poles = pole(feedback(MotorTF,1));
Z = tzero(feedback(MotorTF,1));
PlantZPK = zpk(Z,Poles,1);
PlantZPK = dcgain(feedback(MotorTF,1))/dcgain(PlantZPK) * PlantZPK;
step(PlantZPK);
```

程序执行后,同样产生如图 10.4 所示的曲线。命令 feedback,pole 和 tzero 也适用于这个模型。

10.2.3 离散时间模型

状态空间、传递函数和零点 - 极点 - 增益模型的离散时间形式可以用命令 c2d 产生。在

默认情况下,函数 c2d 采用零阶保持器近似。函数 step,impulse 和 feedback 同样支持离散时间系统模型。当多个单元联合使用时,各个部分的采样时间应该相同。如果在 MATLAB 命令窗口键入:

```
c2d(MotorSS,0.001)
```

将显示下面的系统描述:

```
a =
                  \theta      \omega        i
       \theta      1          0.00099818    0.00018374
       \omega      0          0.99507       0.31535
            i      0          -0.015768     0.36458
b =
                volts
       \theta   1.3203e-05
       \omega   0.036748
            i   0.12617
c =
                  \theta        \omega          i
       \theta      1              0             0
d =
                  volts
       \theta      0
```

注意离散时间模型的矩阵 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 与连续时间模型在本质上是不同的。离散时间模型的状态方程用差分方程描述,而不用微分方程。在状态空间模型中,用 k 表示时刻:

$$x[k+1]=A_d x[k]+B_d u[k]$$

$$y[k]=Cx[k]+Du[k]$$

上式是矩阵的乘法运算。这些矩阵的乘法运算在采样时间 $t=k\Delta$ 与式(10.3)给出的连续时间微分方程的特征相匹配。其中,Δ 是适当选择矩阵 $\boldsymbol{A}_d$ 和 $\boldsymbol{B}_d$ 时的采样间隔,采用零阶保持器近似为[①]:

$$\boldsymbol{A}_d=\mathrm{e}^{\boldsymbol{A}\Delta}$$

$$\boldsymbol{B}_d=\int_0^{\Delta}\mathrm{e}^{\boldsymbol{A}(\Delta-\tau)}\boldsymbol{B}\,\mathrm{d}\tau$$

将得到的左半个复平面的特征值映射到一个单位圆内。在 MATLAB 中还可得到三角近似、双线性近似(Tustin)、改进的双线性近似及匹配近似等多种方法。

在命令窗口键入命令 c2d(MotorTF,0.001)则离散时间系统的多项式描述如下:

```
Transfer function:
1.32e-005 z^2+4.191e-005z+8.023e-006
----------------------------
 z^3-2.36 z^2+1.727 z-0.3678

Sampling time:0.001
```

① 参见 T. Kailith, *Linear Systems Theory*, Prentice Hall, Englewood Cliffs, NJ, 1980 年;或 K. Astrom 和 B. Wittenmark, *Computer Controlled Systems*,第三版, Prentice Hall, Upper Saddle River, NJ, 1997 年。

例 10.3　连续时间模型转换为离散时间模型

将连续时间模型转换为离散时间模型,采样时间为 1 ms。为了明显比较程序的图形输出,在连续时间系统响应与离散时间系统响应的结果之间加了一个很小的偏移,使得两条曲线分开。

```
[ydiscrete,time] = step(feedback(c2d(MotorSS,0.001),1));
ycontinous = step(feedback(MotorSS,1),time);
plot(time,ycontinous,'k-.',time,ydiscrete + 0.1,'k-');
grid;
xlabel('Time'(s)');
ylabel('Response');
legend('Continuous','Discrete',4)
```

程序的结果如图 10.6 所示。由于采样时间很短,所以两条阶跃响应曲线之间没有明显差别。

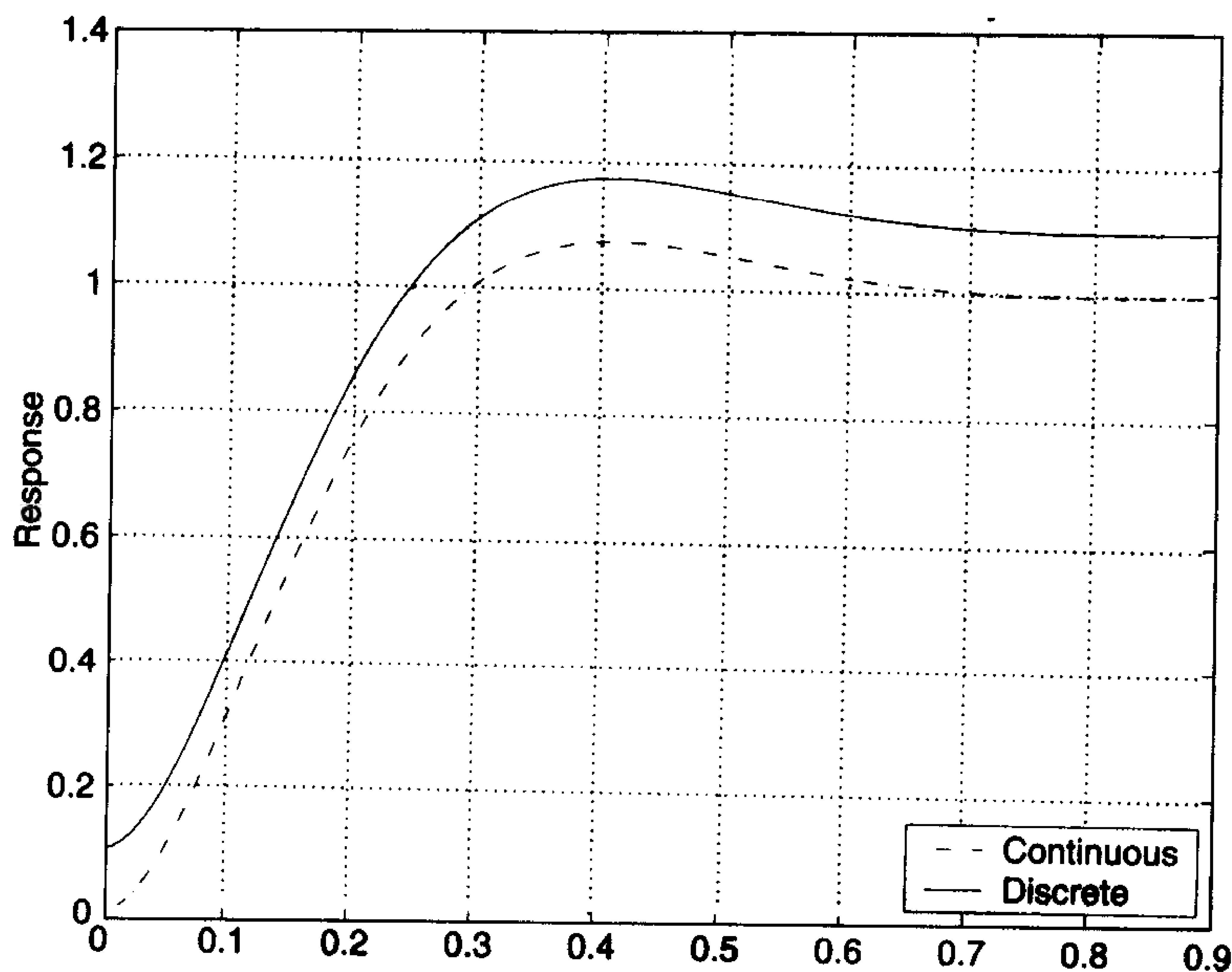


图 10.6　离散时间阶跃响应与连续时间阶跃响应的比较(离散时间响应引入了 0.1 的垂直位移)

在离散时间中,传递函数是用复变量 z 代替复变量 s 的多项式。变量 z 用来描述一个采样时间(周期),代替微分的转换。考虑一个典型的离散时间传递函数:

$$H(z) = \frac{Y(z)}{R(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0} \tag{10.7}$$

式(10.7)两边同乘 $R(z)$,可得:

$$a_n z^n Y(z) + a_{n-1} z^{n-1} Y(z) + \cdots + a_0 Y(z)$$
$$= b_m z^m R(z) + b_{m-1} z^{m-1} R(z) + \cdots + b_0 R(z)$$

转换为离散时间后,方程如下:

$$a_n y[j+n] + a_{n-1} y[j+n-1] + \cdots + a_0 y[j]$$
$$= b_m r[j+m] + b_{m-1} r[j+m-1] + \cdots + b_0 r[j]$$

令 $k=j+n$ 和 $p=n-m$,求解 $y[k]$,则:

$$y[k]=\frac{1}{a_{n-1}}(-a_{n-1}y[k-1]-\cdots-a_0y[k-n]+b_mr[k-p]$$
$$+b_{m-1}r[k-p-1]+\cdots+b_0r[k-p-m]) \quad (10.8)$$

嵌入式系统内核常包含一个控制和滤波算法计算单元,实现如式(10.8)所示的计算过程,即无限脉冲响应滤波器(IIR,infinite impulse response)。在每个采样周期,计算单元依据上一次输出值及本次输入值进行一系列的乘加运算以实现这一算法。用于嵌入式系统和数字滤波器的专用计算机可方便地实现类似的操作。

10.2.4 方框图和仿真工具 SIMULINK

典型的控制系统由几个不同的单元组成,如对象单元和控制器单元,等等。许多复杂结构的系统包括很多子系统,这些输入/输出之间的连接图由系统设计者用方框图设计,基本元件组成如图 10.7 所示。MATLAB 提供了内置函数和运算符集合用于计算传递函数,其过程与方框图的代数运算类似。MATLAB 使用标准运算符,包括 +,-,*,/,并且还提供 feedback,series 和 connect 等函数,以便在命令行中执行和仿真方框图。用户可以利用 SIMULINK 工具箱提供的图形用户界面直接设计方框图,并且仿真特性曲线。

如图 10.7(a)所示,由乘法运算符确定两个系统的级联。例如,若系统对象 H1 和 H2 分别由传递函数 $H_1(s)$和 $H_2(s)$描述,则级联系统 $H_3(s)$可以由下面的等式获得:

```
H3 = H1 * H2
```

除法也得到支持,但要求模型是严格正则的,在这里不进行讨论。

如图 10.7(b)所示,可以用加法和减法运算符得到两个系统的并联。若系统 $H1$ 和 $H2$ 并联,则最后的系统为:

```
H3 = H1 + H2
```

减法的情况可以通过改变上面等式的符号获得。

对于传递函数,图 10.7(c)所示的反馈连接不同于其他两种运算,执行过程不是用运算符实现而是用函数实现的。若 $H4$ 和 $H5$ 反馈连接,则最后闭环系统为:

```
H3 = feedback(H4,H5);
```

以上假设为负反馈。若希望是正反馈,则 $H_3(s)$由下式得到:

```
H3 = feedback(H4 ,H5, + 1);
```

典型串联反馈控制结构如图 10.1 所示,是由图 10.7(a)的串联形式和图 10.7(c)的反馈形式连接而成。在这种情况下,输入减去对象(H_2)的输出,形成误差信号反馈送给控制器(H_1)。下面建立闭环模型,设:

```
H4 = Controller * Plant
```

和

```
H5 = 1
```

则闭环系统 *clSys* 为：

```
clSys = feedback(Controller * Plant,1);
```

通过向量输入/输出方式，函数 feedback 和 series 及操作符 * , + 和 - 支持 MIMO 系统。此外，矩阵运算允许快速构建 MIMO 系统。例如，双入单出系统 *H*3，可由 SISO 系统 *H*1 和 *H*2 来创建，用下式实现：

```
H3 = [H1,H2]
```

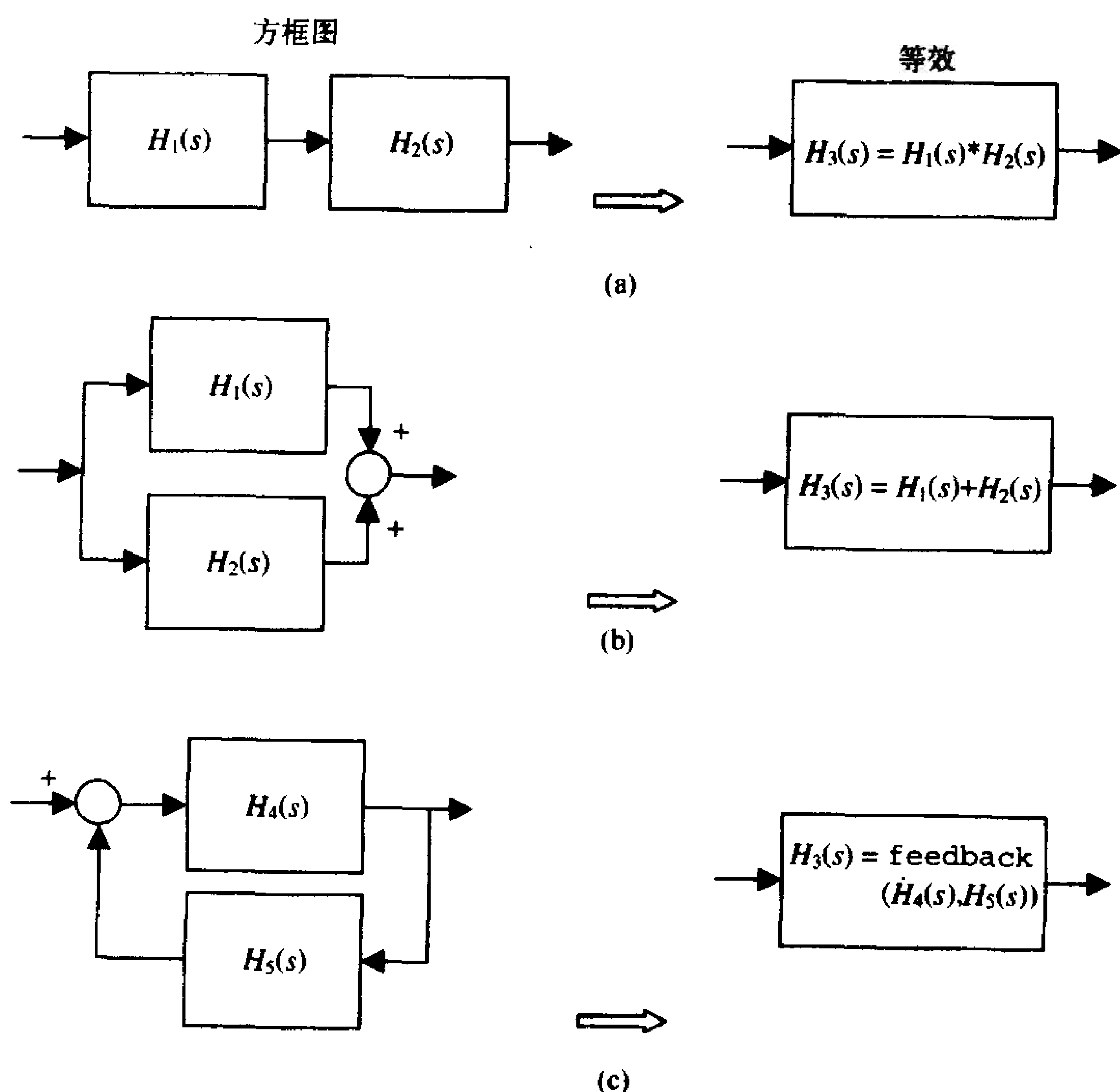


图 10.7 最常用的方框图代数运算

单入双出系统可以这样来创建：

```
H3 = [H1;H2]
```

子系统 *H*1 和 *H*2 可以用 2.4 节讨论的获得子矩阵的方法从 *H*3 中获取。在 MATLAB 系统中，串联、并联和反馈连接要求明确规定谁的输出连到谁的输入上去。函数 series 和 parallel 能够满足这个要求。许多复杂 MIMO 系统可以用 ift 和 connect 来创建。本章仅研究 SISO 控制系统。

若系统对象 *H*1 和 *H*2 的类型不同，则必须进行类型转换，以保证最后模型是统一的。在所有模型中，MATLAB 首先支持状态空间模型。因此，计算过程中只要有一个模型为状态空间模型，则结果就为状态空间形式。对于传递函数和零点 - 极点 - 增益模型而言，MATLAB 先支持传递函数模型。

SIMULINK

SIMULINK 允许用户从元件库建模，并且对系统进行仿真。从元件库中选中元件并拖到模型窗口，连接各个不同的元件并设置每个元件参数，就可以运行模型并显示结果。

SIMULINK 可以通过在 MATLAB 命令中键入下面的命令产生：

```
simulink
```

MATLAB 命令窗口中定义的变量可以在 SIMULINK 窗口中获得。

下面举例说明如何用 SIMULINK 建立直流伺服电机的模型，最后的结果如图 10.8 所示。为了产生对象模型，先在 MATLAB 命令窗口中定义并获得系统矩阵，然后打开 SIMULINK，即：

```
[A,B,C,D] = ssdata(MotorSS);
simulink
```

打开 SIMULINK 浏览窗口，显示四个主要的元件库：*Simulink*, *Control System Toolbox*, *Neural Network Blockset* 和 *Simulink Extras*。首先，点击空白页标记打开 SIMULINK 模型窗口，然后再返回浏览窗口双击 SIMULINK，显示 SIMULINK 元件库的目录：

Continuous
Discontinuities
Discrete
Look Up Tables
Math Operations
Model Verification
Model Wide Utilities
Ports and Subsystems
Signal Attributes
Signal Routing
Sinks
Sources
User-Defined Functions

若想打开任意库中的元件(模块)，只须双击需要的元件库即可。先打开 *Continuous* 元件库。接着用鼠标单击 *State-Space* 模块，并按住键不放，将该元件拖到模型窗口。将元件放在期望的位置并放开鼠标。下一步，打开 *Math* 元件库，依次选中 *Slider Gain* 和 *Sum* 模块，并将其拖到模型窗口，都放在 *State-Space* 模块的左边。然后打开 *Source* 元件库选择 *Step* 模块，并将其放在模型窗口中所有模块的左边。最后，从 *Sinks* 元件库选择 *Scope* 模块，将其放在所有模块的右边。

下一步要做的是设置模块的参数。从 *State-Space* 模块开始，双击该模块，将得到 *State-Space* 模块参数窗口，有五个需要输入数据的地方：A, B, C, D 和 *Initial Condition*。因为在 MATLAB 命令窗口中已经定义了这些矩阵，所以分别键入 A, B, C 和 D 即可。如果选择 MATLAB 命令窗口中定义的不同变量名，则应该在相应的位置输入这些变量名。设初始状态为 0，可以采用同样的方式改变加号，第二个符号应改为负号。变增益模块的增益值是可以选择的，选择范围为 0～15，本例中选为 1。对 *Step* 模块，除 *Step Time*(时间偏移或延迟)选为 0 外，其他为默认值。双击 *Scope* 模块，得到一个模拟的示波器，这时不需要对其进行参数设置。

连接模块形成反馈控制系统。设计窗口中的模块可以用鼠标移动，也可以修改其大小，每

个模块外面的凸起部分都是用来连接的。默认情况下,输入在左边,输出在右边。利用鼠标从输出端拖到输入端即可连成一条线。将十字准线放在 *Step* 模块的输出位置,并按住鼠标左键不放,将十字准线移到 *Sum* 模块的加法(+)输入端。重复这种操作,直到将方框图连成为止,如图 10.8 所示。*State-Space* 模块和 *Scope* 模块中间一点到求和模块减法输入端的连线可以这样产生。将箭头放在两个模块之间的连线上,同时按下 *Ctrl* 键和鼠标左键,保持这种操作并将十字准线移到求和模块的减法输入端,放开鼠标和 *Ctrl* 键。若要调整连线,简单地单击连线并按住鼠标左键,即可将其移到需要的位置。

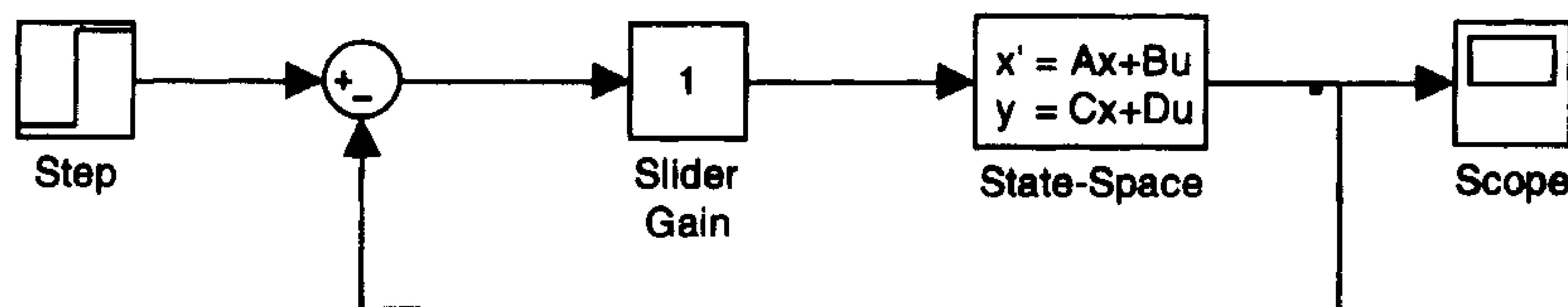


图 10.8 在 SIMULINK 中创建直流伺服电机框图

模块的位置和大小并不重要,关键是连接。在创建连接之前,需要将一些模块进行反转,使得输入在左边,输出在右边。这种操作可以这样完成:先选中模块,然后在下拉菜单 *Format* 内选取 *Flip Block* 即可。另一种方式是在格式菜单中隐藏模块下面的注释,选中模块并选取 *Hide Name* 即可隐藏注释。若要再显示模块注释,选中模块并选取 *Show Name* 即可。但这两种选择不能同时出现。

若要进行仿真,在 *Simulation* 菜单中选取 *Start*,仿真结束后,双击 *Scope* 就可以看到仿真结果。用 x 轴和 y 轴放大图标可以调整图像清晰度。注意在 *Simulation* 菜单下,可以修改仿真参数。例如,可以通过修改参数(选择 *Parameters*)改变综合方法。如果图形连接得不完全或不正确,则仿真不能进行,在 MATLAB 命令窗口会给出错误信息,并弹出一个特殊窗口。对图 10.8 所示的仿真模型在 *Scope* 中的结果如图 10.4 所示。

10.2.5 描述方法的转换

MATLAB 通过构造函数 ss,tf 和 zpk 进行三种描述之间的转换。例如,在 tf 中,用状态空间模型 **MotorSS** 产生传递函数模型。

```
PlantTF = tf(MotorSS)
```

转换过程中数值误差将产生微小的系统模型系数差异。

函数 ss 可以将传递函数转换为状态空间模型,因此:

```
PlantTF = tf(MotorSS);
PlantSS = ss(PlantTF)
```

转换后的系统矩阵 $\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 与 **MotorSS** 定义的矩阵不同。这说明,对于给定的系统,其状态空间模型描述不是惟一的,这与传递函数模型不一样。一般使用:

```
ssbal
```

来调节输入量、状态量和输出量,使得仿真条件尽可能好。因此,用:

```
PlantTF = tf(MotorSS);
PlantSS = ss(PlantTF)
PlantBal = ssbal(PlantSS)
```

试图找到系统的最佳条件。

零点 - 极点 - 增益模型也可以由状态空间模型或传递函数模型转换。例如,零点 - 极点 - 增益模型可以由状态空间模型产生,用:

```
PlantTF = tf(MotorSS);
PlantSS = ss(PlantTF)
PlantZ = zpk(PlantSS)
```

这种转换与其他转换方法一样,在 MATLAB 中使用数值解有时会有很大的数值误差,尤其是对 10 阶以上的系统求解①。在实际情况中,建议不要经常变换描述方法。

10.3 系统响应

本节将介绍一些工具的用法,这些工具用来计算系统响应。适用于三种连续和离散时间描述的计算系统响应的函数包括 step,impulse,initial 和 lsim。在使用方框图时仿真工具箱提供的许多内置信号源使仿真得到了简化。使用函数 arx 可以从系统响应估计系统。

无论是传递函数描述还是状态空间描述,配置零极点都是控制工程师估计系统响应的便捷方法,许多设计要求可以转化为零极点位置的约束条件。控制器的设计问题经常转变为反馈回路的设计,应使得闭环系统极点位于复平面中期望的位置,但是零点的位置不能通过反馈来改变。

10.3.1 根据系统估计响应

在控制系统的响应中,阶跃输入经常用来作为比较不同控制器的基准。MATLAB 函数 step 用于计算阶跃响应,若不需要返回值,则仅画出响应曲线。函数自动给出适当的时间范围来计算仿真,因此程序如下:

```
[theta,t] = step(MotorSS);
plot(t,theta)
xlabel('Time  (s)')
ylabel('\theta(t)')
```

程序执行后,如图 10.9 所示。从图中可以看出,电动机的阶跃响应是一条斜线,因为恒定电压加在线圈上,所以电动机最后会达到恒定的转速。

另一方面,可以不用 step 提供的时间范围而自己指定时间 t 的范围。在这种情况下,需要将前面程序段的第一行改为下面的两行表述:

---

① 参见 N.E.Leonard 和 W.S.Levine, *Using MATLAB to Analyze and Design Control Systems* ,Benjamin/Cummings,Redwood City,CA,1992 年。

```
t=linspace(0,100,100);
theta=step(MotorSS,t);
plot(t,theta)
xlabel('Time(s)')
ylabel('\theta(t)')
```

其中,$0\leqslant t\leqslant 100$。

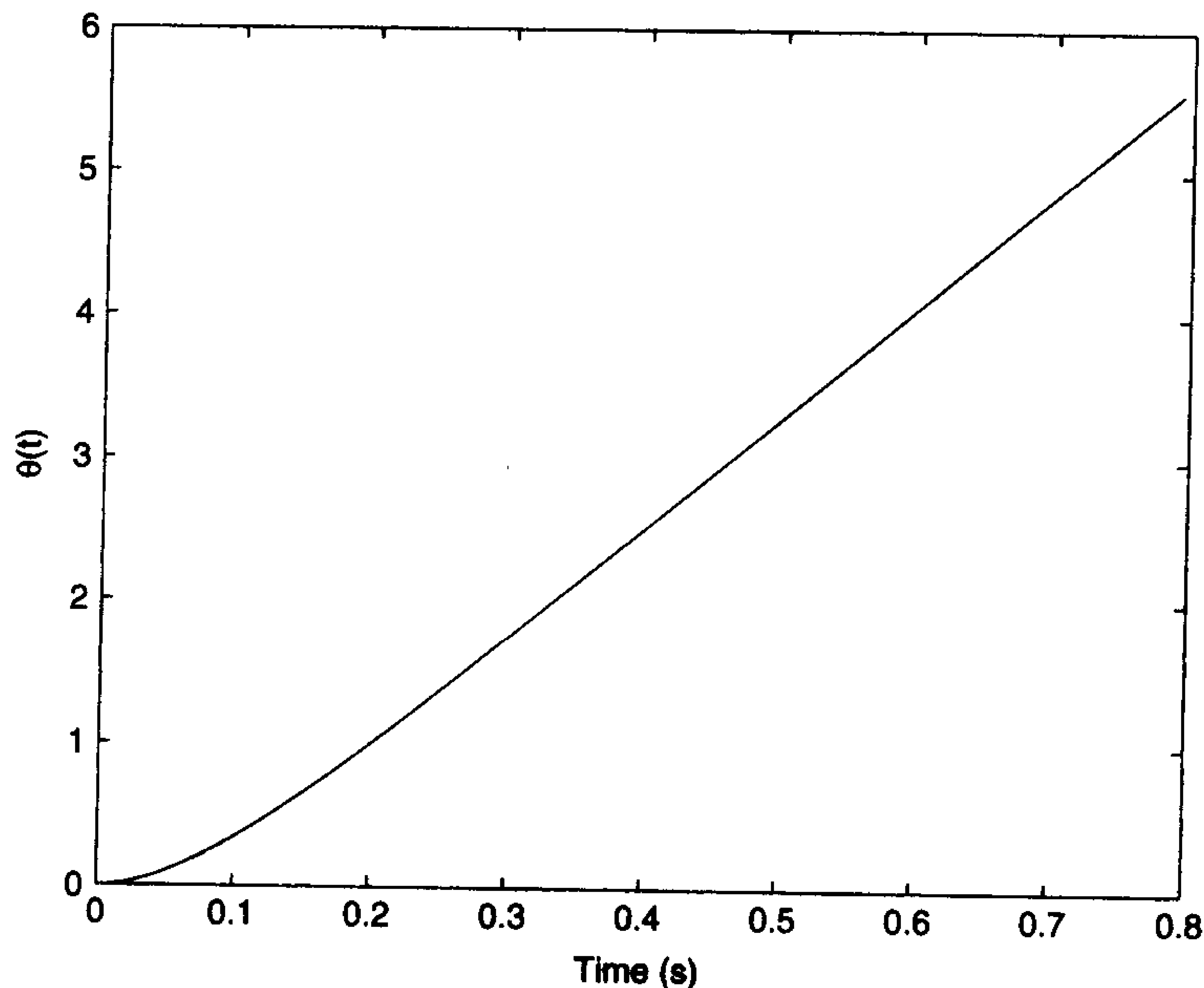


图 10.9 无反馈的电动机阶跃响应曲线

MATLAB 函数 impulse 用于计算系统的脉冲响应,其用法与 step 类似。

MATLAB 提供的函数不能适应于所有的标准检测输入,但是使用下式则提供了在任意输入条件下求解系统响应的能力:

```
lsim(sys,u,t)
```

其中,*sys* 为研究中的系统,而 u 是描述输入幅值的向量,也是时间 t 的函数。u 和 t 的长度必须相等。

例 10.4 电机控制系统的跟踪误差

图 10.3 中将确定电动机控制系统跟踪误差的稳定特性。当输入是斜坡信号时,稳态误差是期望位置与实际位置之间的最终差值。在 $0\leqslant t\leqslant 1$ 的范围内,输入信号斜率为 1。对直流伺服电机而言,当 $0\leqslant t\leqslant 1$ 时,随动误差 $e(t)=\theta(t)-t$,程序为:

```
t=linspace(0,1,100);
theta=lsim(feedback(Motorss,1),t,t);
plot(t,t'-theta,'k-');
error=t(end)-theta(end);
hold on;
plot([0,t(end)],[error,error],'k--');
```

```
xlabel('Time(s)');
ylabel('Error e(t)');
```

执行后产生图 10.10。注意,经过最初的暂态以后,误差近似稳定在 0.13。控制系统设计目标是使这种误差值小于 0.05。

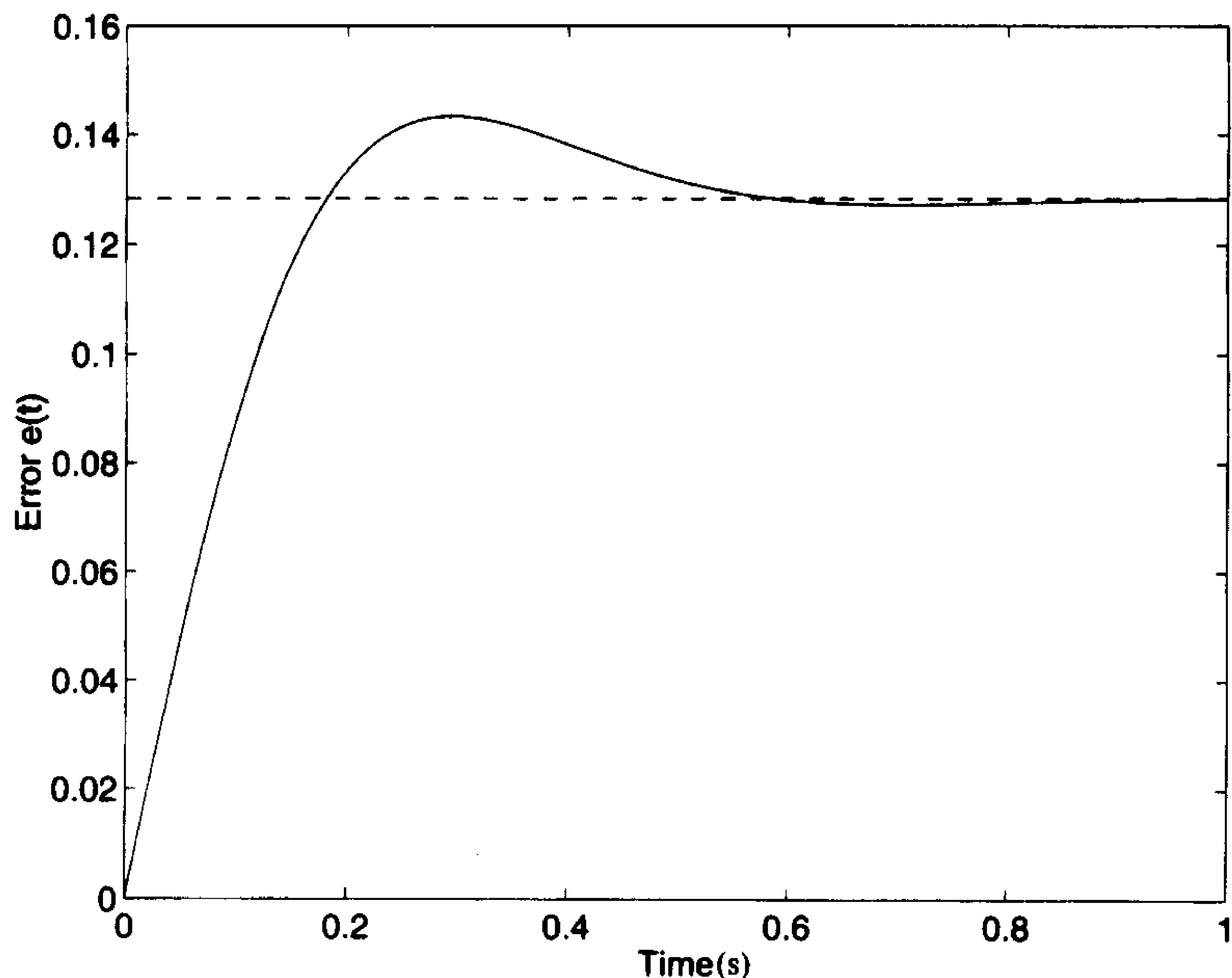


图 10.10 随时间变化的控制系统跟踪误差

对于状态空间系统,MATLAB 函数

```
initial(sys,x0)
```

在非零初始条件下进行仿真,其中 *sys* 是被研究的系统,$x0$ 是初始条件向量。

例 10.5 直流电机在初始条件下的响应

考虑图 10.3 给定的直流电机在初始位置 $\theta(0)=0$,初始电流 $i(0)=0$,初始角速度$\omega(t)=5$ rad/s。此外,还假定增益为 2。若线圈上的电压保持在 0 伏(通过短路实现),则下面的程序将计算电机在这些初始状态下的响应曲线。

```
x0=[0;5;0];
[theta,t,x]=initial(feedback(2*MotorSS,1),x0);
plot(x(:,1),x(:,2)'k-');
grid on;
xlabel('\theta(t) (radians)');
ylabel('\omega(t) (radians/s)');
```

其中,$x(:,1)=\theta(t)$,$x(:,2)=\omega(t)$,$x(:,3)=i(t)$。程序执行后的结果如图 10.11 所示。

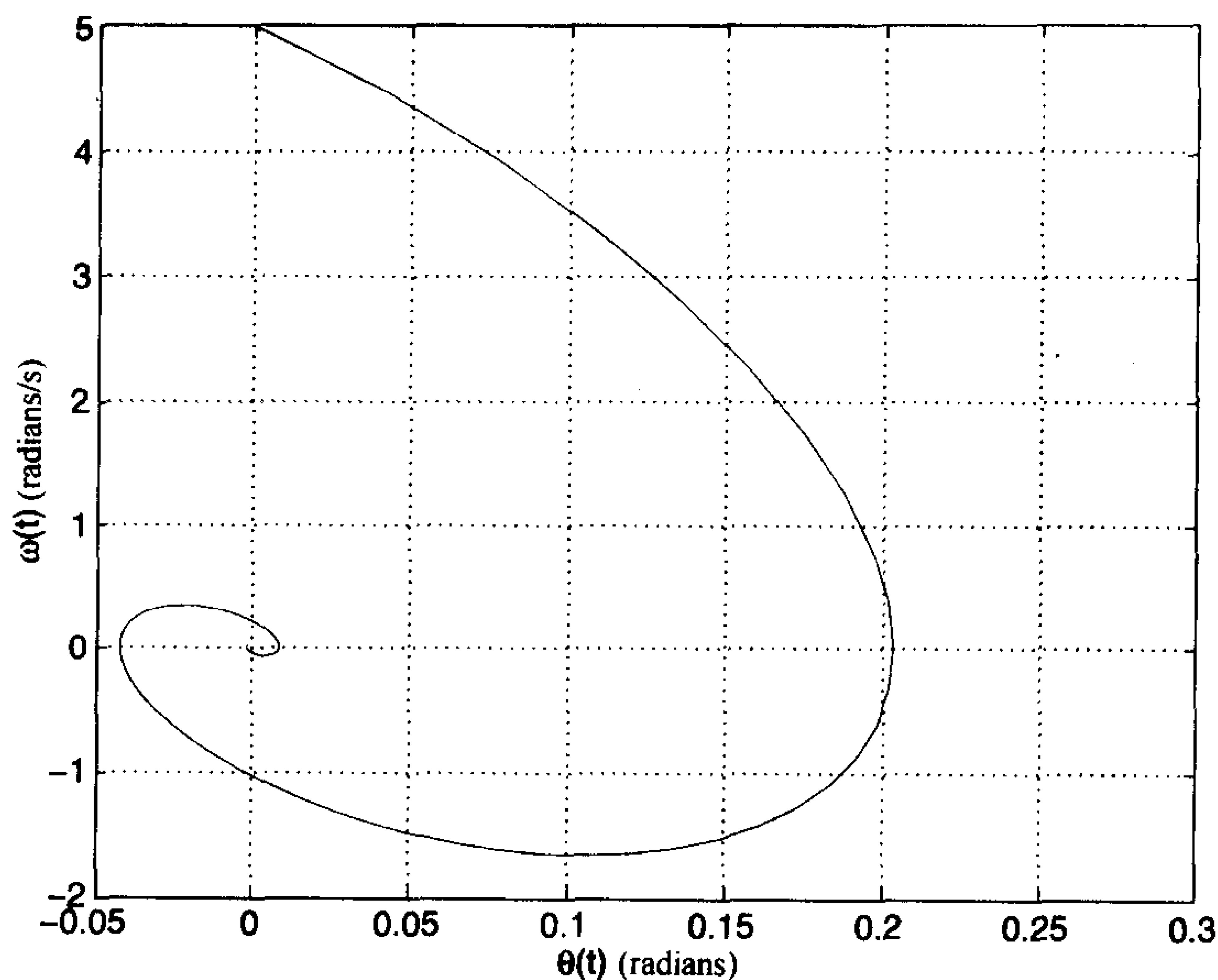


图 10.11 初始角速度 $\omega(t)=5$ rad/s 的转子相位曲线

10.3.2 根据零极点估计系统响应

用 MATLAB 解决控制问题时,重要的是能定性地判断微分方程的解。由于控制器设计是一个反过程,所以,定性判断能让设计者知道为了达到设计要求应采用什么样的系统。MATLAB 通过求解各种类型的方程可以提高对微分方程求解的定性理解程度。

本节将计算许多系统的阶跃响应,这些系统具有不同的零极点。首先,研究具有一个极点没有零点的一阶系统。假设极点位置为 $-\sigma$,则传递函数为:

$$H(s)=\frac{\sigma}{s+\sigma}$$

分子设为 σ 是为了保证系统的直流增益为 1。下例说明如何获得系统阶跃响应。

例 10.6 具有不同极点的一阶系统阶跃响应

极点由慢至快从 $\sigma=0.1$ 变化到 $\sigma=2$,产生一系列阶跃响应曲线。

程序如下:

```
t=0:0.1:10;
polevect=0.1:0.1:2;
hold on;
for i=1:length(polevect)
  y=step(tf([polevect(i)],[1,polevect(i)]),t);
  polt(t,y,'k-');
end;
xlabel('Time');
ylabel('Step response');
text(0.2,0.95,'p=2');
text(3,0.1,'p=0.1 ( \ Deltap=0.1)')
```

程序执行后的结果如图 10.12 所示。

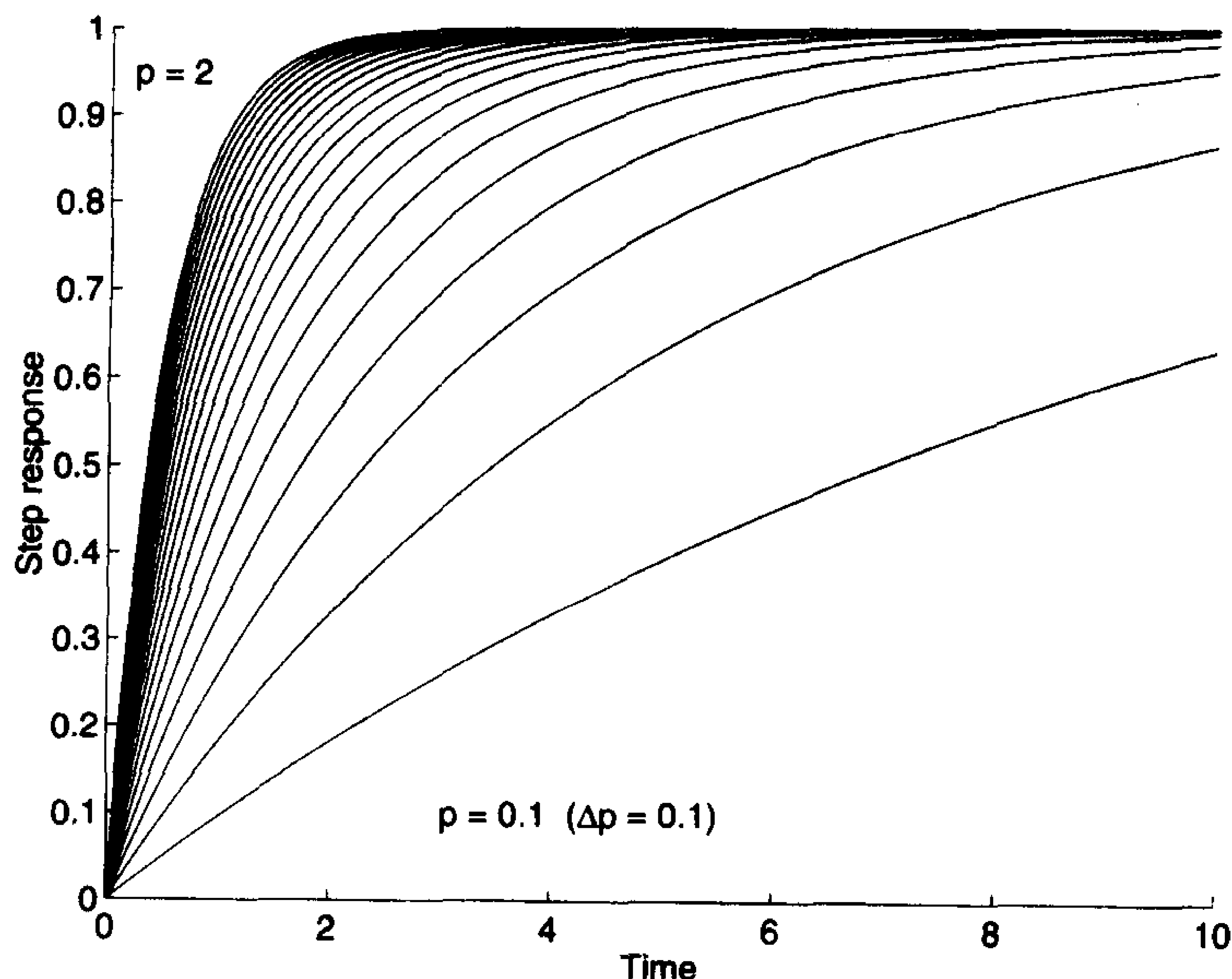


图 10.12 极点 p 变化情况下的一阶系统的响应曲线

当一阶系统的极点接近虚轴(即 σ 变得很小)时,控制系统会变得非常迟钝。通常这种迟钝对控制系统来说并不好,但有时缓慢响应能抑制一些扰动,这种迟钝反而变成有利的一面。因此,系统极点的最佳位置取决于控制系统的设计目标。

若将两个一阶系统串联起来,则形成了一个二阶系统。许多机械系统都是二阶的,因此掌握好二阶系统是非常重要的。研究下面的二阶系统:

$$H(s)=\frac{\sigma^2+\omega^2}{s^2+2s\sigma+\sigma^2+\omega^2} \tag{10.9}$$

系统极点为 $-\sigma\pm j\omega$,其中 σ 是极点的实部,而 ω 是极点的虚部。通常,σ 表示系统的衰减,ω 为机械储能或弹性。因为在此关心的是系统响应与极点的依赖关系,所以传递函数被极点的位置参数化。下面根据不同极点位置计算式(10.9)。

例 10.7 具有不同极点的二阶系统阶跃响应

研究式(10.9)所描述系统对不同极点的响应。

首先,设假 $\omega=1.0$,以 σ 为变量,画出阶跃响应;然后再设 $\sigma=0.5$,以 ω 为变量,画出阶跃响应。而且在每一个响应旁边画出了极点的轨迹。

```
t=0:0.4:10;sigma=linspace(0.05,1.0,10);
data=zeros(length(t),length(sigma));
omega=1.0;
for i=1:length(sigma)
 data(:,i)=step(tf[sigma(i)^2+omega^2],...
                   [1,2*sigma(i)sigma(i)^2+omega^2]),t);
end;
subplot(2,2,1);
mesh(t,-sigma,data');
ylabel('\sigma');
xlabel('Time');
```

```
zlabel('Response');
title('Response as a function of \sigma: \omega = 1.0');
data = zeros(length(t),length(omega));
sigma = 0.5;omega = linspace(0.3,2.0,10);
for i = 1:length(omega)
 data(:,i) = step(tf([sigma^2 + omega(i)^2],[1 2 *sigma sigma^2 + omega(i)^2]),t);
end;
subplot(2,2,3);
mesh(t,omega,data');
ylabel('\omega');
xlabel('Time');
zlabel('Response');
title('Response as a function of \omega: \sigma = 0.5');
subplot(2,2,2);
hold on;
plot([-0.1,-0.1],[1.0,-1],'x');
plot([-0.1,-1],[1.0,1.0]);
plot([-0.1,-1],[-1.0,-1.0]);
plot([-1,-1],[1.0,-1],'<');
plot([-2,1],[0,0],'k')
plot([0,0],[-2,2],'k');
axis([-2,1,-2,2]);
xlabel('Real axis');
ylabel('Imaginary axis');
title('Pole location');
subplot(2,2,4);
hold on;
plot([-0.5,-0.5],[0.3,-0.3],'x');
plot([-0.5,-0.5],[0.3,1.5]);
plot([-0.5,-0.5],[-0.3,-1.5]);
plot(-0.5,1.5,'^');
plot(-0.5,-1.5,'v');
plot([-2,1],[0,0],'k');
plot([0,0],[-2,2],'k');
axis([-2,1,-2,2]);
xlabel('Real axis');
ylabel('Imaginary axis');
title('Pole location');
```

程序执行后，结果如图10.13所示。

像一阶系统那样，稳态响应为 $H(0)=1$。如图10.13所示，当 σ 固定，而 ω 增长时，系统响应衰减很小。当 ω 固定，而 σ 增长时，系统响应衰减很大。如果极点的实部接近虚轴，则系统衰减也很小。如果极点进入复平面的右半平面，则系统响应是无限大的。

电机系统是高阶系统，一般具有二阶系统特性，有时也具有一阶系统特性。在系统极点中，如果有一些极点的实部比其附近的极点实部大很多，且附近没有传输零点，则这些极点将决定系统的输入/输出特性。这样就允许设计者用一阶或二阶系统近似表示系统的闭环特性。为理解这些，要研究直流永磁电机的速度控制问题。运动方程与式(10.2)相同，只是输出用转子的角速度 $d\theta(t)/dt=\omega(t)$ 代替了角度 $\theta(t)$。程序如下：

```
PlantRPM = MotorSS;
PlantRPM.c = [0,1,0];
```

```
PlantRPM = minreal(PlantRPM);
set(PlantRPM,'OutputName','\omega');
pole(PlantRPM)/(2*pi)
```

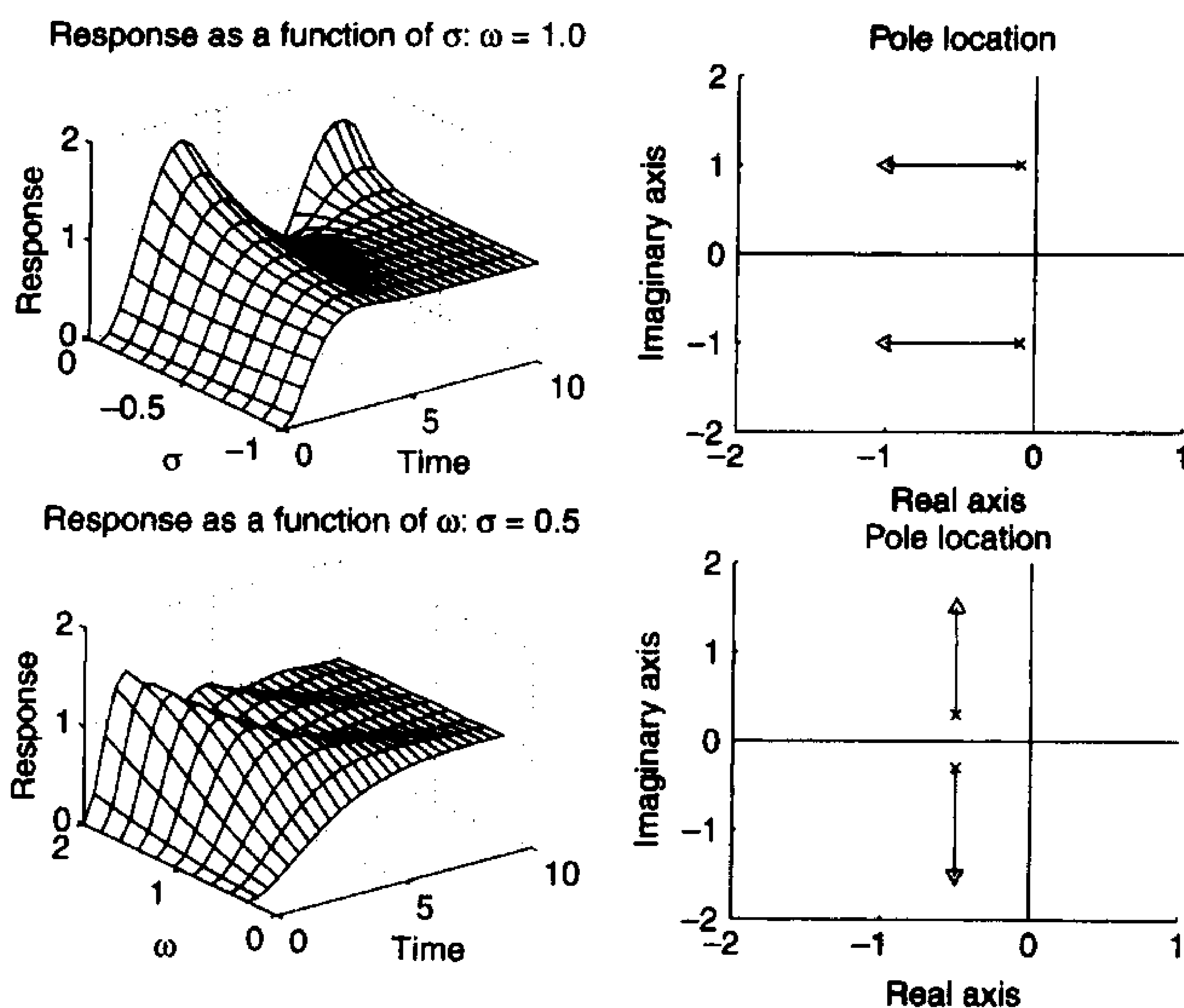


图 10.13　极点位置变化的二阶系统响应曲线

程序的第一行创建系统 *PlantRPM*,然后读出矩阵 *C* 确定 $\omega(t)$ 而不是 $\theta(t)$。由于转子角度 $\theta(t)$ 已经是不可观测的,所以调用 minreal 从方程中消去了这个状态量。

最后一行程序返回系统的极点:一个快速极点约在 160 Hz 位置,与线圈有关;另一个约在 2 Hz 位置,是慢速极点,与转子运动有关。因为快速极点比慢速极点的速度高 10 倍还多,所以由转子运动而引起的慢速极点决定系统的开环响应。设转子速度 $\omega(t)$ 和线圈电流 i 都在 -1 和 $+1$ 之间变化,在此条件下程序进行仿真,并将结果绘在同一个相位曲线图中,即 $\omega(t)$ 与 $i(t)$ 的曲线图。时间没有明显标在相位图上,所以当全部轨线时间超过 5% 时,在轨线上标出一个小圆圈。

```
PlantRPM = MotorSS;
PlantRPM.c = [0,1,0];
PlantRPM = minreal(PlantRPM);
set(PlantRPM,'OutputName','\omega');
t = linspace(0,0.05,500);
hold on;
for i = 0:13;
 x0 = [-1.4+0.2*i,1];
 [y,t,x] = initial(PlantRPM,x0,t);
 plot(x(:,2),x(:,1),'k-');
 k = floor(0.05*length(t));
 plot(x(k,2),x(k,1),'ro');
end;
for i = 0:13;
 x0 = [-1+0.2*i,-1];
 [y,t,x] = initial(PlantRPM,x0,t);
 plot(x(:,2),x(:,1),'k-');
```

```
 k = floor(0.05 * length(t));
 plot(x(k,2),x(k,1),'ro');
end;
plot([1,-1],[0,0],'k--');
plot([0 0],[,1 -1],'k--');
axis([-1,1,-1,1]);
ylabel('i(t)(A)');
xlabel('\omega(t) (radians/s)');
```

系统的相位图如图 10.14 所示。注意，每条曲线的大部分时间处于一维子空间。这个子空间与慢速极点有关系。通常，这些实部很大的极点决定系统的阶跃响应。

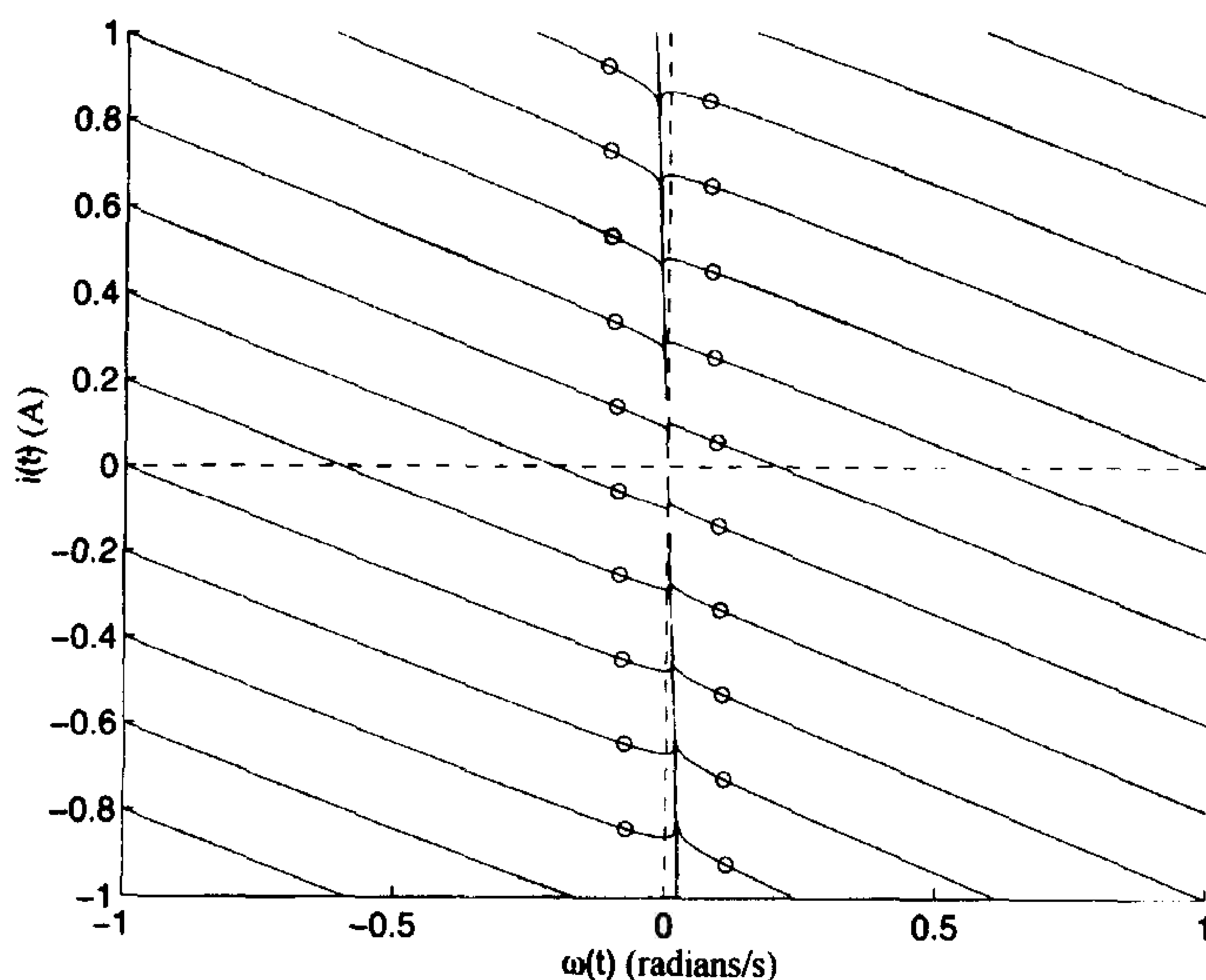


图 10.14　直流电机的相位图

例 10.8　极点附近的零点，对二阶系统的影响

研究二阶系统：

$$H(s)=\frac{-(s-z)}{z(s^2+0.5s+1)}$$

其中，z 是零点位置，尽管零点位置不能通过反馈来改变，但是如果这些零点在极点附近或虚轴附近，那么其位置影响还是很重要的。为了说明这一点，研究一类对象的阶跃响应。其零点有的靠近虚轴，有的穿越虚轴。当 $z=-5$ 和 $z=-1$ 时，零点是稳定的；然而在 $z=1$，$z=5$ 的情况下，零点是不稳定的。给 z 选定四个值：± 5 和 ± 1。程序为：

```
t = linspace (0,25,200);
Z = [-5,-1,1,5];
Den = [1,0.5,1];
y = zeros(length(t),length(Z));
for i = 1:length(Z)
 y(:,i) = step(-1/Z(i) * tf([1 -Z(i)],Den),t);
end
plot(t,y(:,1),'k-')
hold on
```

```
plot(t,y(:,2),'k--')
plot(t,y(:,3),'k-o')
plot(t,y(:,4),'k-+')
legend('zero at -5','zero at -1','zero at 1','zero at 5')
xlabel('Time')
ylabel('Step response')
```

程序执行结果如图 10.15。

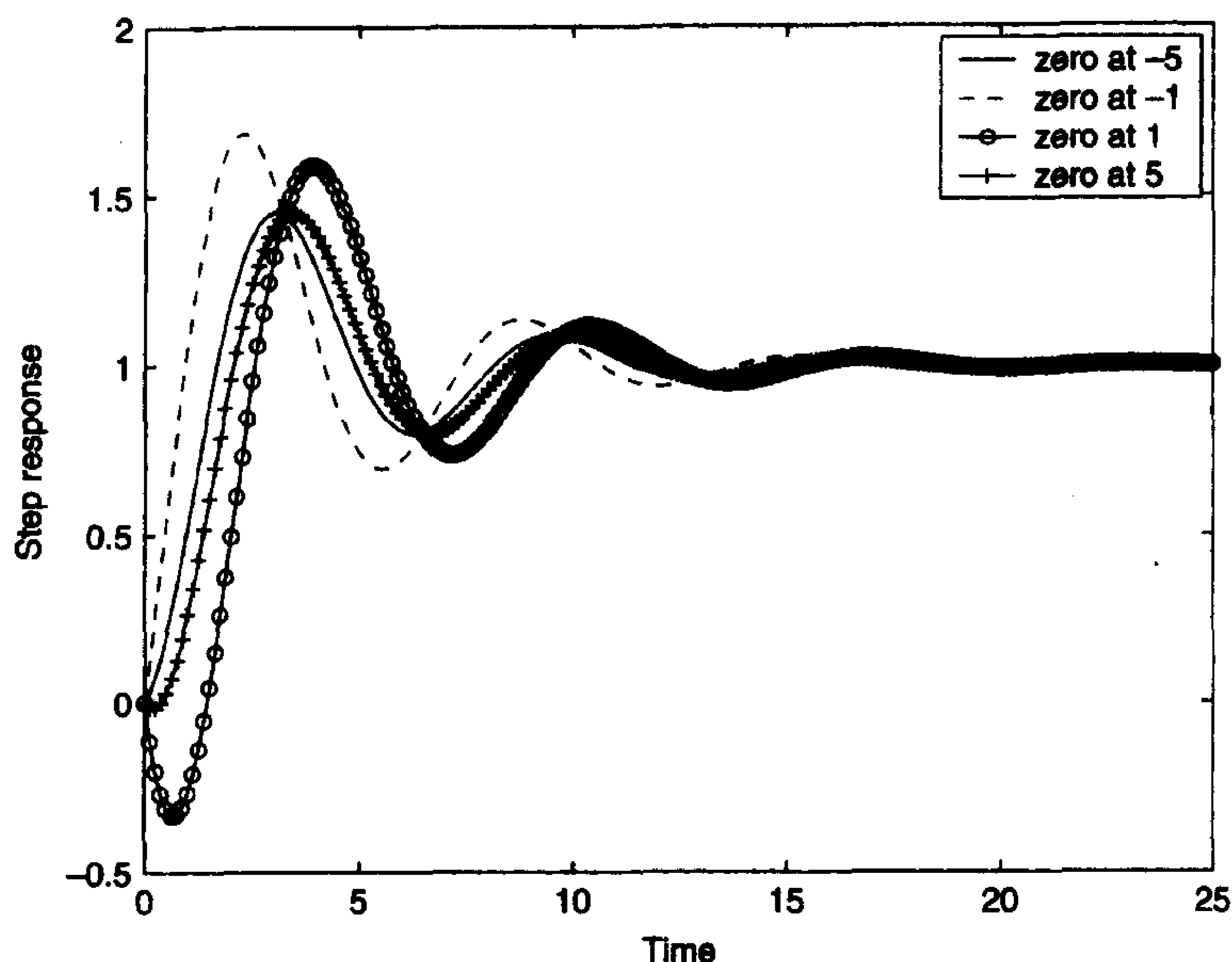


图 10.15 临近和穿越虚轴零点产生的影响

当 $z=-5$ 和 $z=-1$ 时，零点是稳定的；然而在 $z=1, z=5$ 的情况下，零点是不稳定的。尽管不稳定零点不会使系统不稳定，但是限制了反馈量。如图中位置为 1 的零点是不稳定零点，会使系统性能恶化。具有一个或多个不稳定零点的系统称为非最小相位系统。

例 10.9 形式动态特性的屏蔽

研究下式给出的四种不同阶次的系统。

$$G_1=\frac{1}{s+1} \qquad G_3=G_1(s)*G_2(s)$$

$$G_2=\frac{100}{s^2+10s+100} \qquad G_4=\frac{19.8s+20}{s+20}G_3(s)$$

对象 G_3 的阶跃响应与上述的一阶对象 G_1 的响应非常类似，因为极点 -1 为决定性复极点。但是，若在极点 -1 附近有一个零点，则极点将不起决定性作用。由 G_4 确定的系统在 G_3 极点 -1 附近增加的零点掩盖了其作用。然而这在实际中一般不能做到，因为要使 G_3 快速响应，则 G_4 首先应产生很大的输出，这可能使执行器饱和或损坏。从 G_4 的阶跃响应中可以看到这一点。通过下面的程序可得到 G_3 和 G_4 的阶跃响应曲线，如图 10.16 所示：

```
G1 = tf([1],[1,1]);
G2 = tf([100],[1,10,100]);
G3 = G1 * G2;
G4 = tf([19.8,20],[1,20]) * G3;
t = linspace(0,6,200);
yG3 = step(G3,t);
yG4 = step(G4,t);
plot(t,yG3,'k--',t,yG4,'k-')
legend('G3','G4')
xlabel('Time')
ylabel('Response')
```

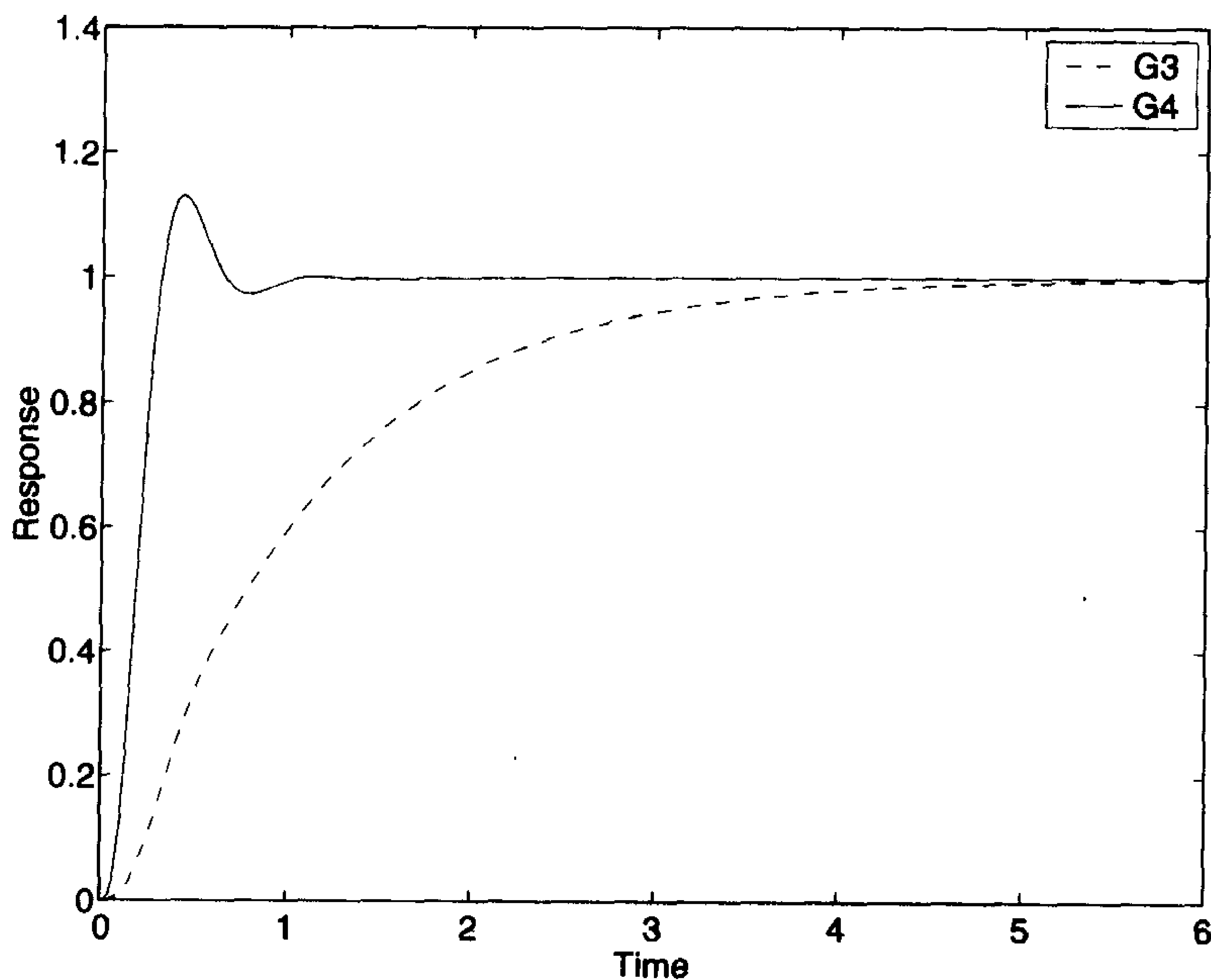


图 10.16 零点隐藏了系统的缓慢动态特性

利用零点和极点之间的距离可以度量输入或输出矩阵与特征向量的正交性。如果零点在极点的正上方，那么它的模在输出端是不可激励的或不可见的。所谓能控性和能观性就是用于描述这种现象的。

10.3.3 由响应估计系统

本节将介绍系统辨识技术。在 MATLAB 中，系统辨识工具箱提供了若干系统辨识工具，可通过 ident 命令对其进行访问。系统设计过程中，首先应确定系统的原理模型，如本章应用的电动机模型。然后在控制运算法则执行过程中，根据实验数据进行模型的评估并揭示系统的非预期现象，如电动机轴转动足够缓慢时与控制回路由于相互作用而产生的共振。为此，采用外部输入(arx)的自回归模型进行参数辨识。

为在应用反馈控制的电动机模型系统中获得实验数据，创建系统在离散时间域的随机输入量并求出其响应曲线。输入量应具有足够的激励作用以获得有效的辨识。对所有线性系统

而言,其输入量视同零输入①。实验生成一系列输入量 $u[k]$(命令)及响应量 $y[k]$(角度)。采样周期为 T_s。实际上,由于噪声和不可测量的干扰量对信号所造成的污染,数据集很少是完全清洁的。为此,在系统辨识工具箱中使用 iddata 将数据引入辨识数据对象,并将其分为辨识数据集和确认数据集。结果如图 10.17 所示。图 10.18 创建了三个不同角度的 ARX 模型并分别给出了其响应曲线。为实现 ARX 估计功能,要求用户必须给出多项式分子和分母的阶次以及迟滞时间。本例中给出了模拟数据,因此可预知系统具有三个极点,但实际上这是不确定的。detrend 命令用于设定输出和输入相对于零点的均值以满足 ARX 技术初始条件和线性化的要求。三个模型的结果通过系统辨识工具箱中的函数 compare 进行比较。

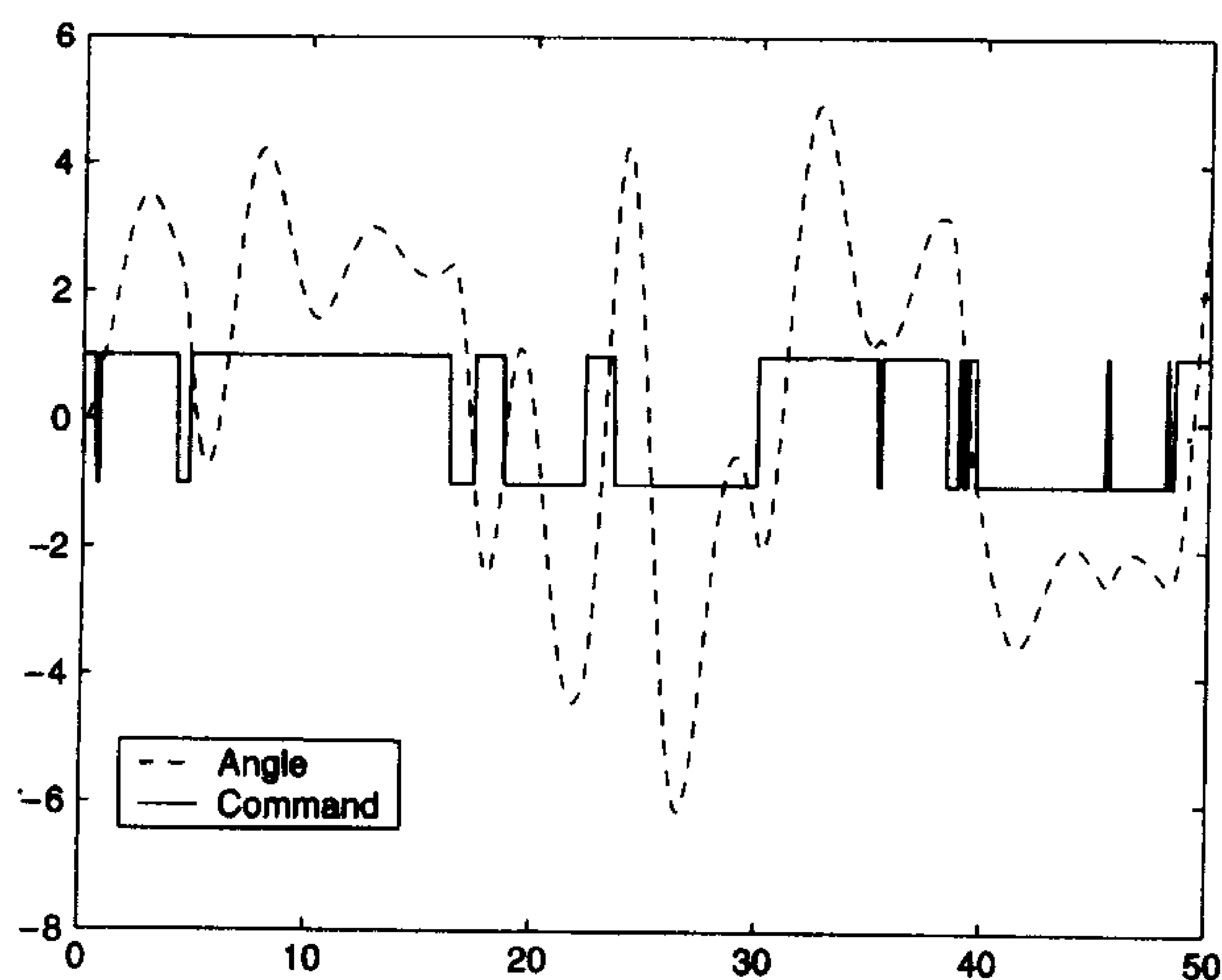


图 10.17 系统辨识数据集

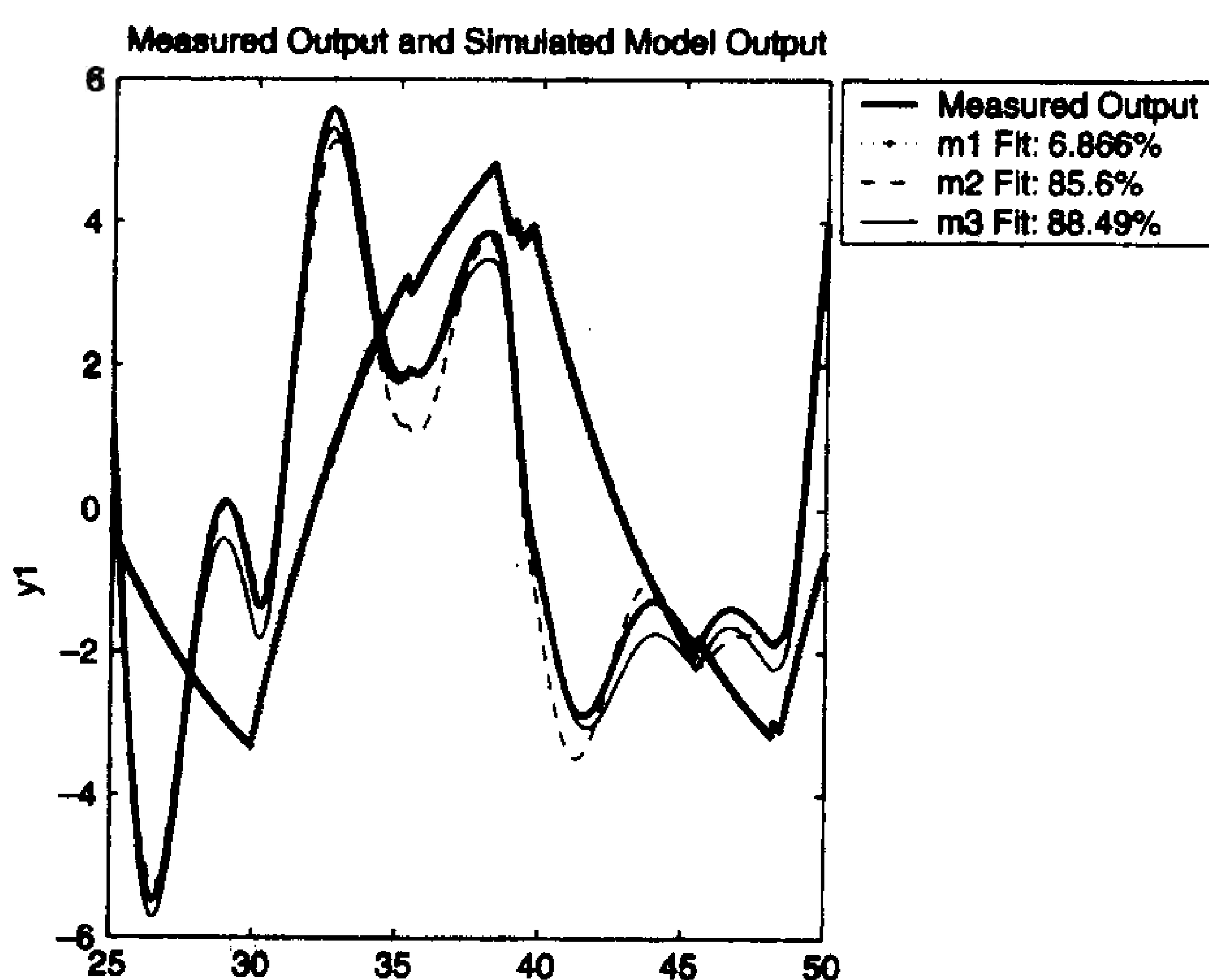


图 10.18 三个系统响应与确认数据集的比较

① 参见 L. Lyung, *System Identification: Theory for the User*,第二版,Prentice Hall, Upper Saddle River, NJ, 1999 年。

```
Ts = 5e - 2;N = 1000;
u = zeros(N,1);
Jmodel = rand;
clSys = feedback(c2d(MotorSS(Jmodel),Ts),0.4);

r = 1;
for i = 1:length(u)
  if rand < 2.5e - 2
    r = - 1 * r;
  end
  u(i) = r;
end
t = Ts * (1:N);
y = lsim(clSys,u);
figure(1)
plot(t,y,'k--',t,u,'k-')
legend('Angle','Command','Location','South West')
motorExp = iddata(y,u,Ts);
motorId = detrend(motorExp(1:N/2));
motorVal = detrend(motorExp(N/2:N));
m1 = arx(motorId,[1,1,0]);
m2 = arx(motorId,[2,1,0]);
m3 = arx(motorId,[3,1,0]);
figure(2)
compare(motorVal,m1,m2,m3)
```

可以看出，在二阶模型 $m2$ 和三阶模型 $m3$ 之间只有很少的改善。系统具有三个极点，其中，两个用于确认系统响应而另一个用于系统辨识。

有些参数模型辨识技术在 Box-Jenkins，Output Error 和 Prediction Error 工具箱中实现，其差异在于模型结构和噪声处理方式不同。系统频率响应可利用非参数辨识方法，通过实验数据直接进行估计。

10.4 设计工具

本节讨论 MATLAB 中的设计工具及设计准则。在计算机普及之前，许多设计都是采用图解的方法，图解方法包括：

bode——波特图法
nyquist——奈奎斯特法
rlocus——根轨迹法

还有基于计算机的现代设计方法，如 lqr 和 lqe。

控制系统设计准则包括三个要求：

1. 稳定性
2. 暂态特性
3. 稳态特性

在解微分方程时，显然需要稳定性。分母多项式的每一个根对应着方程的一个解，并且任何一

个具有正实部的根都会使系统响应按指数规律增加。虚轴上的极点使系统处于临界稳定状态。后面两个设计准则必须建立在稳定的条件下才可以进行研究。暂态特性要求观察系统单位阶跃响应的短期行为,而稳态特性为长期行为,主要用于研究下列输入时的随动误差,可能的输入是阶跃、斜坡或极少用到的抛物线。稳定性准则经常在系统中遇到,一些系统开始时是不稳定的,但最终一定会稳定。倒置摆和磁性轴承都属于这类系统的例子。许多系统利用反馈使其成为稳定系统,闭环控制系统的稳定性可以很容易地用闭环系统传递函数分母上的根来检测。任何具有正实部的根都表明闭环系统是不稳定的。

10.4.1　设计准则

本节以直流电机为例说明设计工具,并讨论直流电机闭环系统的稳定性。选用比例控制器,输入期望位置和实际位置的差,并乘以增益。设计者的主要任务是选择比例控制器的增益。下面的程序产生图 10.19,图中表明增益参数从 1 变化到 200 时的右侧极点实部曲线。

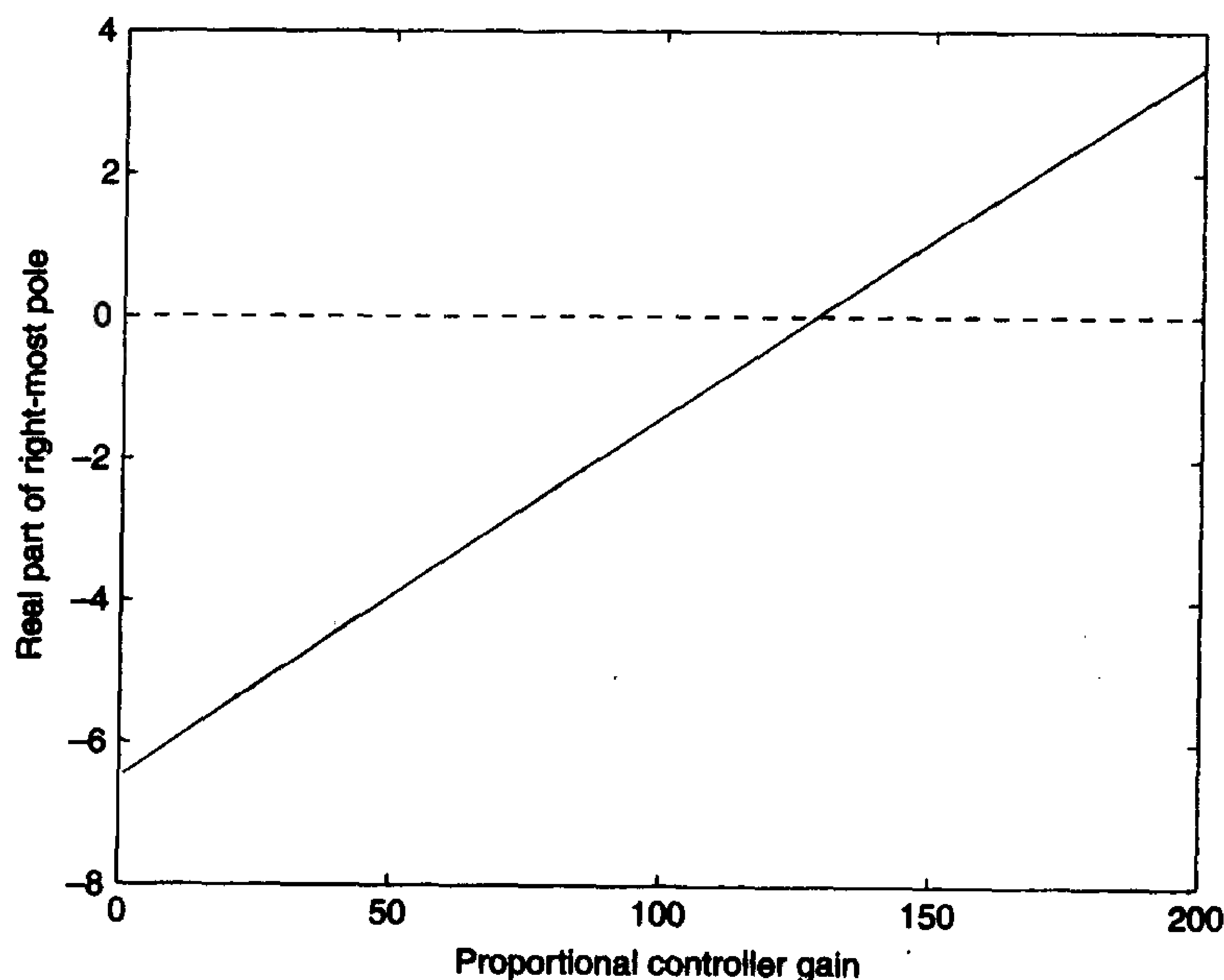


图 10.19　闭环系统右侧极点实部作为控制器增益函数

当增益较小时系统是稳定的;但是当增益远大于 128 时,闭环系统响应将是无限的,此时电机可能被损坏。

```
gains = linspace(1,200,50);y = [];
for i = 1:length(gains)
 y = [y,max(real(pole(feedback(gains(i) * MotorSS,1))))];
end;
plot(gains,y,'k-',[0,200],[0,0],'k--');
xlabel('Proportional controller gain');
ylabel('Real part of right-most pole');
```

由下面的函数产生图 10.18 所示的波特图:

```
bode(MotorSS)
```

并且可确定在多大增益时系统不稳定。命令

```
[gm,pm,wgm,wpm] = margin(MotorSS);
```

用来计算幅值余量(gm)和相角余量(pm)以及与之相对应的频率 wgm 和 wpm。在此,幅值(余量)(在系统变得不稳定之前的幅值)是 gm = 128.37 dB 或 42.2 dB。

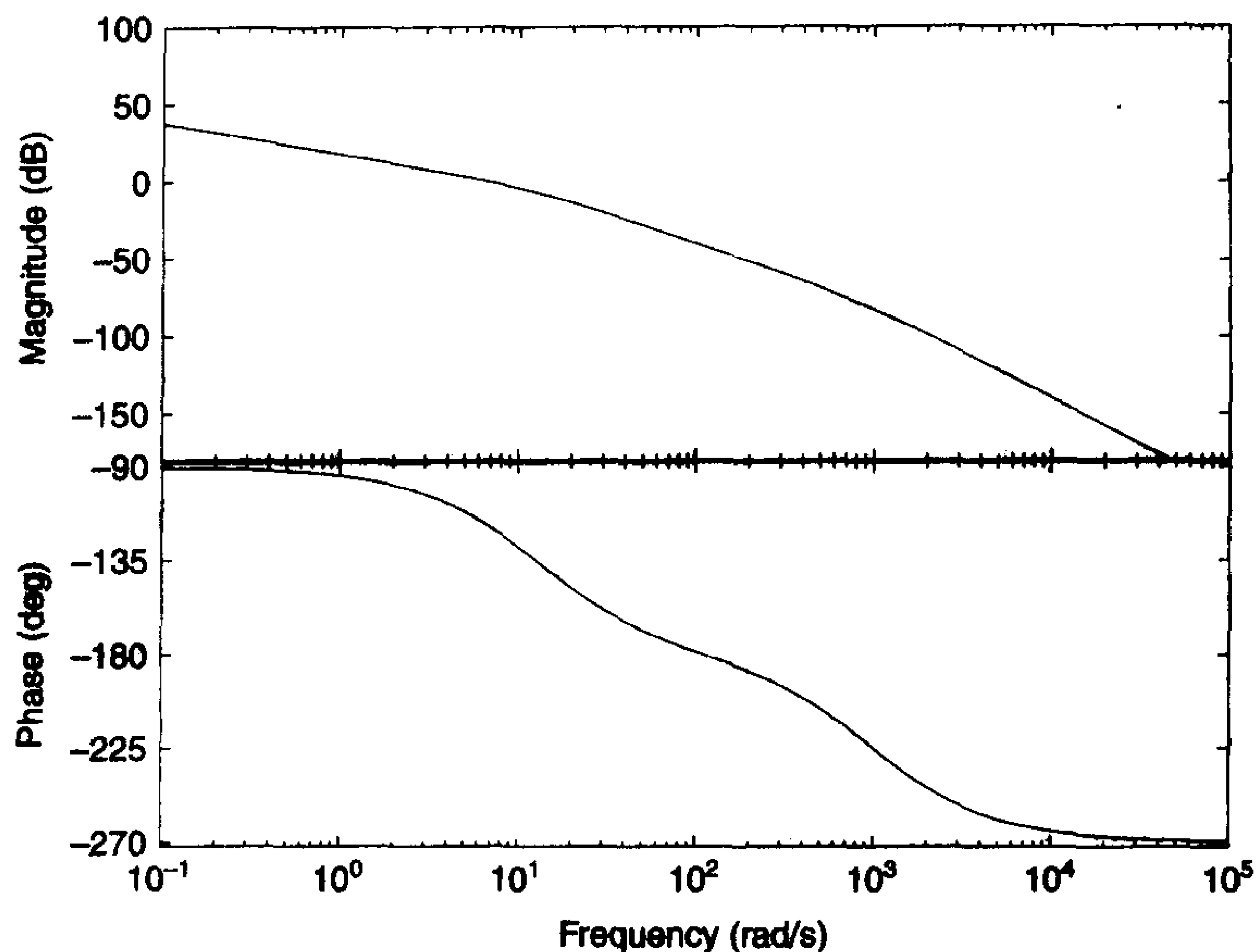


图 10.20 给定相角余量和幅值余量条件下描述截止频率的波特图

即使闭环系统是稳定的,但其闭环特性也可能不令人满意。直流永磁电机系统对一系列稳定增益的阶跃响应曲线如图 10.21 所示,可由下面的程序得到,时间范围是 0 到 1 s。四个增益在 1 到 10 之间按对数比例等间距分布。

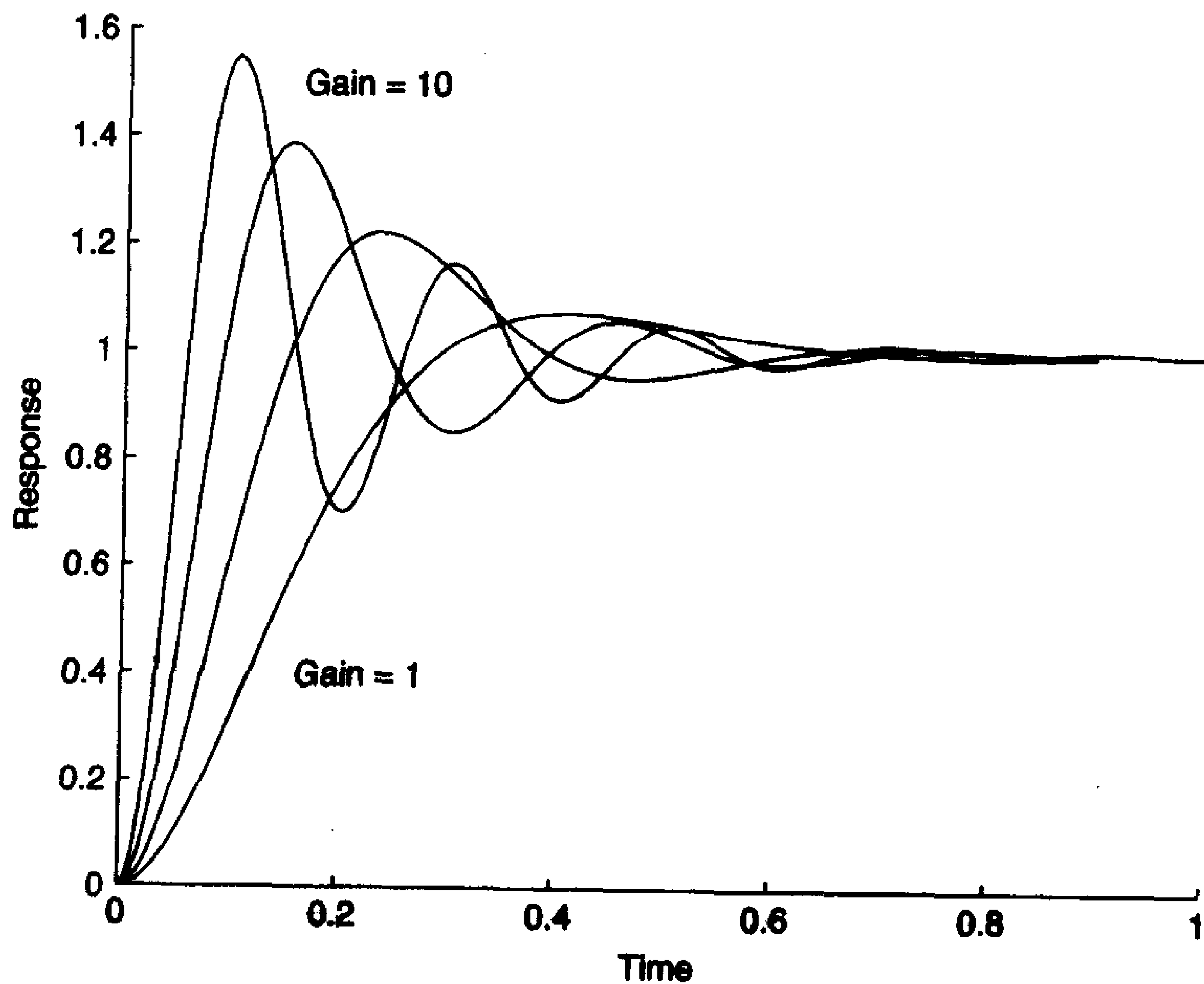


图 10.21 比例控制的电动机系统阶跃响应

```
t=linspace(0,1,100);
gains=logspace(0,1,4);
hold on;
for i=1:length(gains)
  [y,t]=step(feedback(gains(i)*MotorSS,1));
  plot(t,y);
end;
xlabel('Time');
ylabel('Response');
text(0.16,1.5,'Gain=10');
text(0.16,0.4,'Gain=1');
```

随着增益的变大,响应速度加快,高增益的控制器会出现超调,并在预定值附近产生振荡。有很多性能指标可以用来量化这种现象,如上升时间、超调和调节时间等。

暂态特性可以通过系统的阶跃响应测量。上升时间是由终值的10%上升到90%时系统所需的时间。与之相关的值是峰值时间,即系统第一次达到峰值的时间。在特定系统中,上升时间与峰值时间在设计上与超调是相互矛盾的。超调量是指系统超过目标值的大小。调节时间一般定义为系统从开始到进入偏离终值2%的范围内所用的时间。下面的 **transient** 函数计算这些参数。若系统不稳定,则函数返回[-1,-1,-1]。

```
function criteria=transient(system)
criteria=[-1 -1 -1];
maxP=max(real(pole(system)));
if maxP>=0
 return
end
MaxTime=-6*(1/maxP);
Time=linspace(0,MaxTime,500);
Response=step(system,Time);
[ResponseMax,IndexMax]=max(Response);
FinalValue=Response(end);
TimeLow=interp1(Response(1:IndexMax),Time(1:IndexMax),
                  0.1*FinalValue);
TimeHigh=interp1(Response(1:IndexMax),Time(1:IndexMax),
                  0.9*FinalValue);
criteria(1)=TimeHigh-TimeLow;
k=length(Time);
while(k>0)&(0.02>abs((FinalValue-Response(k)/FinalValue));
 k=k-1;
end;
criteria(2)=Time(k);
criteria(3)=100*(max(Response)-FinalValue)/FinalValue;
```

其中,*criteria*(1)=上升时间,*criteria*(2)=调节时间,*criteria*(3)=超调量。

用这个函数估算直流电动机的控制器,调用:

```
transient(feedback(MotorSS,1))
```

显示向量[0.1935, 0.6080,7.6136]。其中,0.1935是上升时间,0.6080是调节时间,而7.6136是超调量。

10.4.2 设计方法

对于 10.4.1 节讨论的设计准则,将介绍一些设计方法,并举例说明这些方法在电动机控制中的应用。常用的三种不同方法是:

1. 基于频率的方法
2. 根轨迹法
3. 基于 LQG 的方法

基于频率的设计方法不需要精确的模型,只需要一批实验结果。其他两种方法都需要精确的模型。

下面举例说明这些设计方法。

例 10.10 满足上升时间和超调百分比准则的控制器设计

现在研究电动机控制器问题。设计准则要求超调量保持在 20%以下,因此图 10.19 所示的许多增益条件都是不被采用的。此外,还要求闭环系统能快速响应,上升时间小于 0.05 s。若直接用比例控制器,将会出现问题。一方面,为了使上升时间小于 0.05 s,增益必须远大于 3;另一方面,为使超调小于 20%,增益必须远小于 2。下面的程序产生图 10.22,图中显示了对于比例控制器超调量和上升时间随增益变化的曲线。

```
kp = 0.4 * logspace(0,1,20);
result = [];
for i = 1:length(kp)
 result = [result;transient(feedback(kp(i) * MotorSS,1))];
end;
[ax,h1,h2] = plotyy(kp,result(:,3),kp,result(:,1));
xlabel('Controller gain');
ylabel('Percentage overshoot');
v = axis;
set(get(ax(2),'Ylabel'),'String','Rise time(s)')
set(h2,'Marker','ks');
set(h1,'Marker','ko');
```

首先采用基于频率的设计方法。超调的指标要求可以转化为相角余量的指标要求。对这个系统而言,相角余量为 45°时,即可满足超调量要求。从图 10.20 的波特图中可以看出,当相角为 -135°(相角余量为 45°)时,允许的增益近似为 2.0。用瞬态下的 fzero 函数更精确地计算增益。程序为:

```
function Gain
options = optimset('display','off');
gain = fzero(@PEcontrol,2,options)
transresp = transient(feedback(gain * MotorSS,1))

function s = PEcontrol(gain)
rval = transient(feedback(gain * MotorSS,1));
s = rval(3)-20;
```

程序执行后,得到超调量为 20%条件下的增益为 *gain* = 1.9384, *transresp*(1) = 上升时间:0.1111 s,上升时间比设计指标慢一倍多。

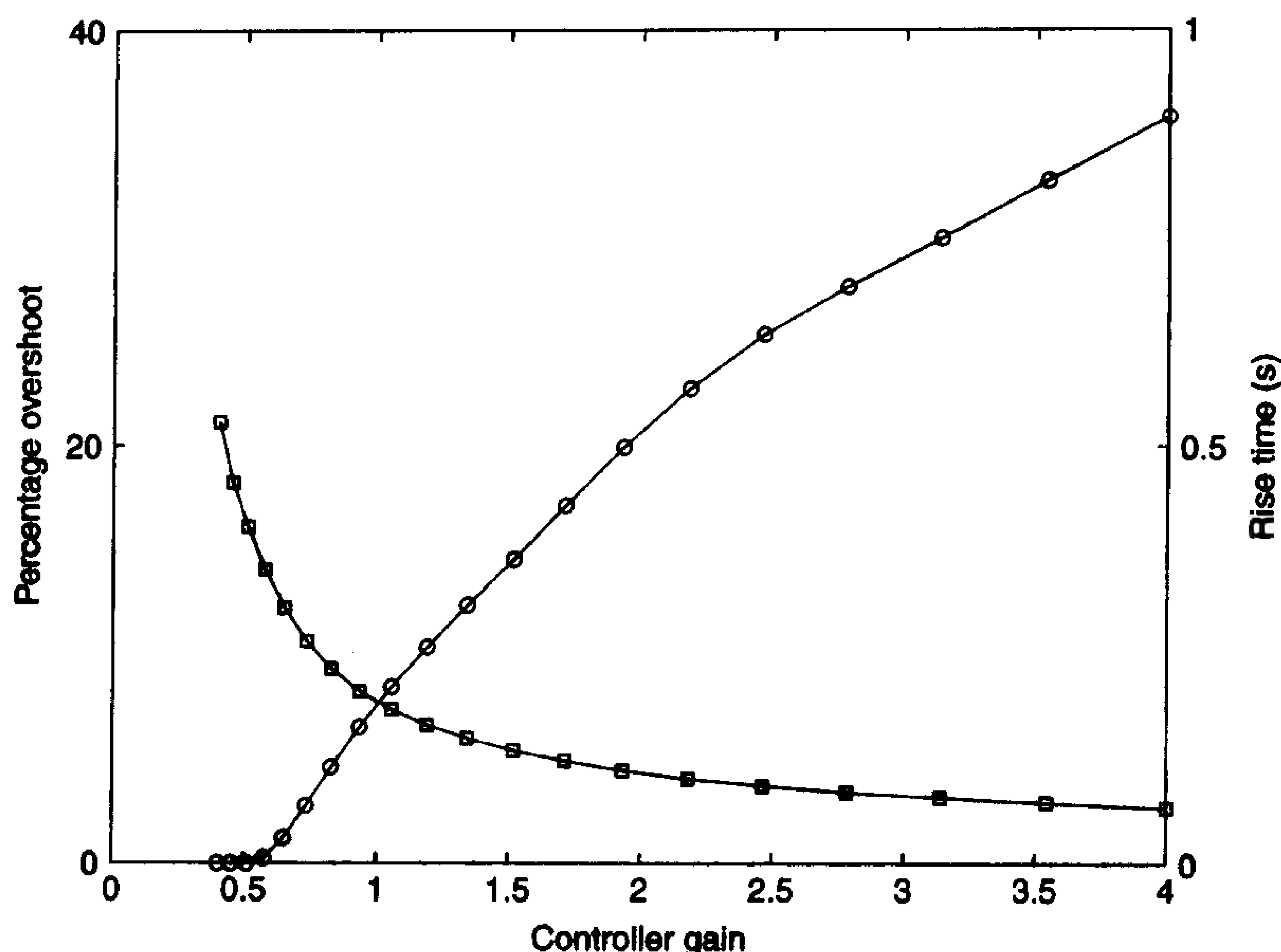


图 10.22 控制器增益函数的超调量(圆圈)和上升时间(方块)曲线

一般采用超前控制器来提高系统的暂态响应,适当地设计超前控制器能在小的频率范围内增加相位;同时,相位增加允许使用更大的增益。超前控制器的零点选在 -15,刚好在第二个开环极点 -13 的左边。这样能确保超前控制器相位增加,与直流电动机相位下降处于同一个位置。超前控制器的极点在 -100,几乎是零点位置的 10 倍。理论上极点位置越靠左越好,但是极点太靠左会使控制器对噪声很敏感。经验法则是极点位置不能超过零点位置的 10 倍。最后的控制器传递函数是:

$$C(s) = \frac{100}{15}\frac{s+15}{s+100} = \frac{6.667s+100}{s+100}$$

将传递函数乘以 100/15,使直流增益为 1。

下面的程序产生图 10.23,用来比较未补偿系统和补偿系统的频率响应:

```
Control = tf([6.667 100],[1 100]);
bode(MotorSS,'k-',Control * MotorSS,'k--')
```

注意高频时相位下降是如何滞后的。使用波特图找到恰当的反馈增益近似为 10。为了计算超调量为 20%时的精确增益,采用下面的程序:

```
function Gain2
options = optimset('display','off');
gain = fzero(@LDcontrol,[10],options)
transresp = transient(feedback(13.0108 * tf([6.667,100],[1,100]) * MotorSS,1))

function s = LDcontrol(gain)
rval = transient(feedback(gain * tf([6.667,100],[1,100]) * MotorSS,1));
s = rval(3)-20;
```

程序执行后,得到超调量为 20%时的 *gain* = 13.0108, *transresp*(1) = 上升时间 ≈ 0.017 s,上升时间比期望值几乎快 3 倍。比例和超前控制的系统阶跃响应曲线如图 10.26 所示。

用超前补偿器可以同时满足超调和上升时间的要求。基于频率的设计不依靠精确的模型

而是从波特图中得到数据。其他基于频率的设计方法还包括奈奎斯特曲线和尼科尔斯(Nichols)图,使用方法与上述方法类似。

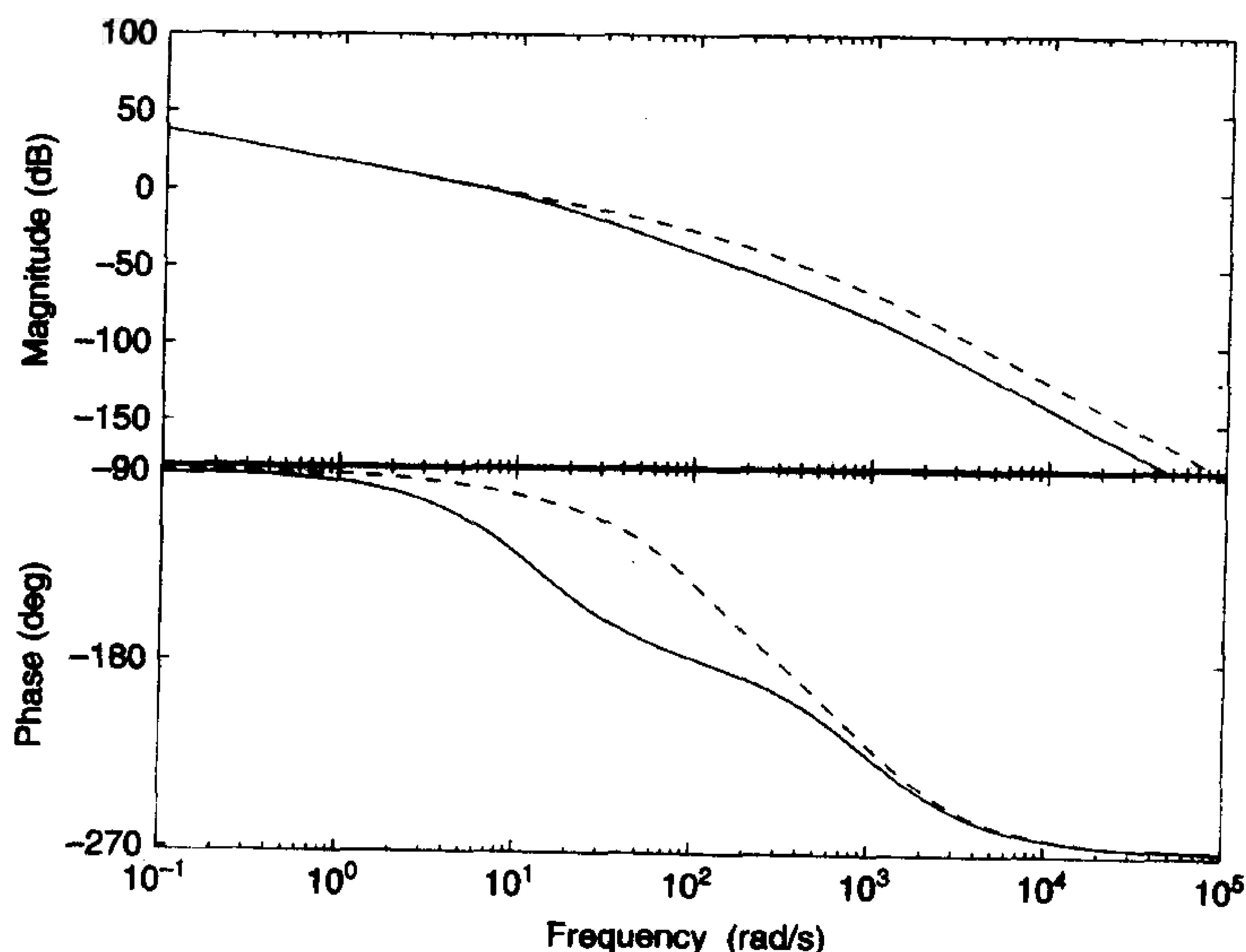


图 10.23 超前补偿系统(虚线)和未补偿系统(实线)的波特图

根轨迹是另一种常用的设计方法,同样是用超前补偿但方法不同。给定复数极点所对应的角度 ξ 及半径 ω_n,为[1]:

$$M_{\mathrm{p}} = \exp\left(-\frac{\pi\xi}{\sqrt{1-\xi^2}}\right)$$

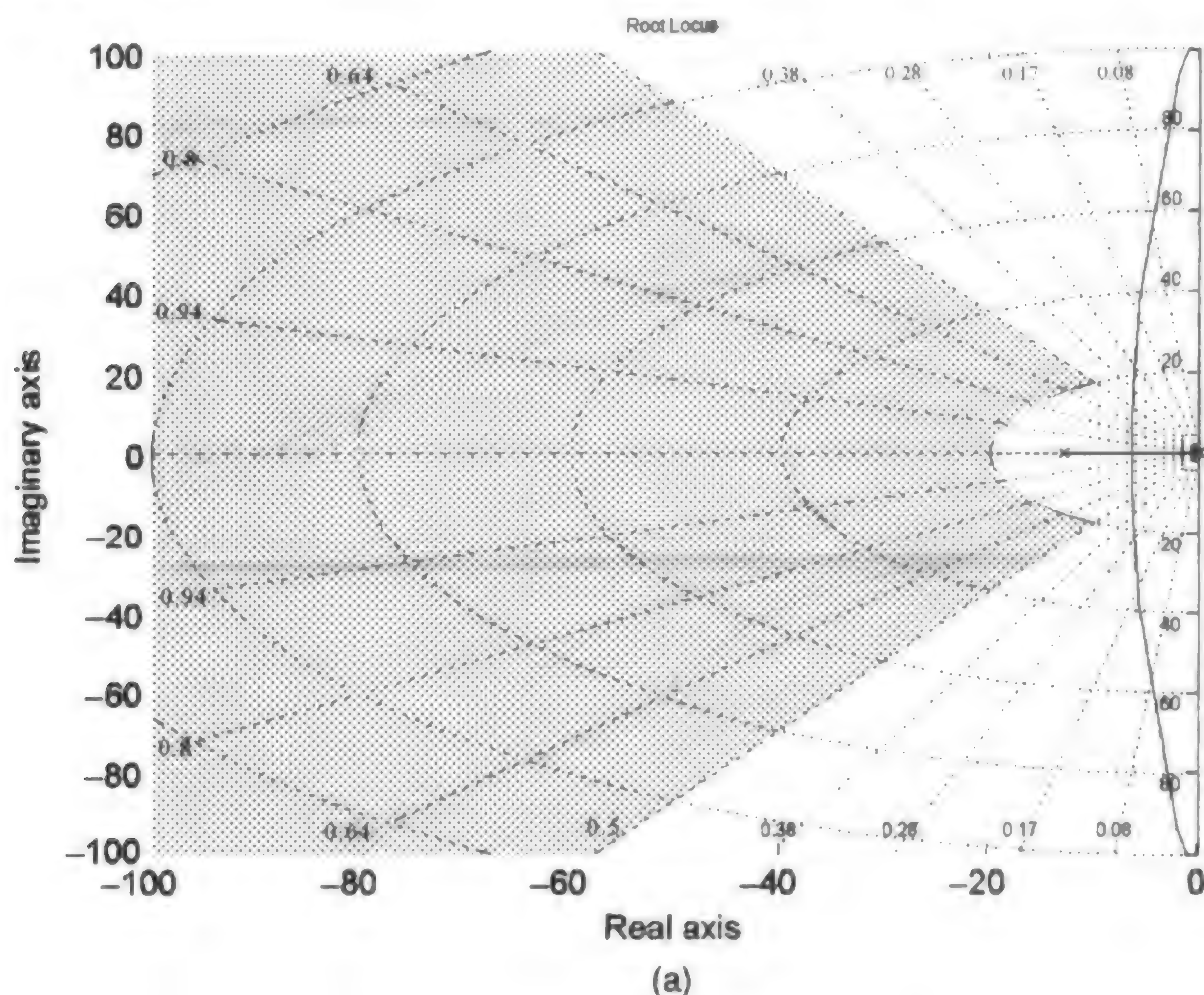
$$T_{\mathrm{r}} \approx \frac{1}{\omega_n}(1+1.4\xi)$$

其中,M_p 为峰值,T_r 为上升时间。转换这些约束条件以满足动态设计要求,并将闭环极点配置在复平面中。超调量小于20%的设计要求限制了 ξ,要求主导极点位于以负虚轴为中心的120°楔形内。上升时间0.05 s大约对应最小的极点半径 $\omega_n = 20$。这些都是经验法则,起着很重要的作用。下面的程序首先确定闭环极点在复平面上的区域,然后画根轨迹检验是否满足条件。

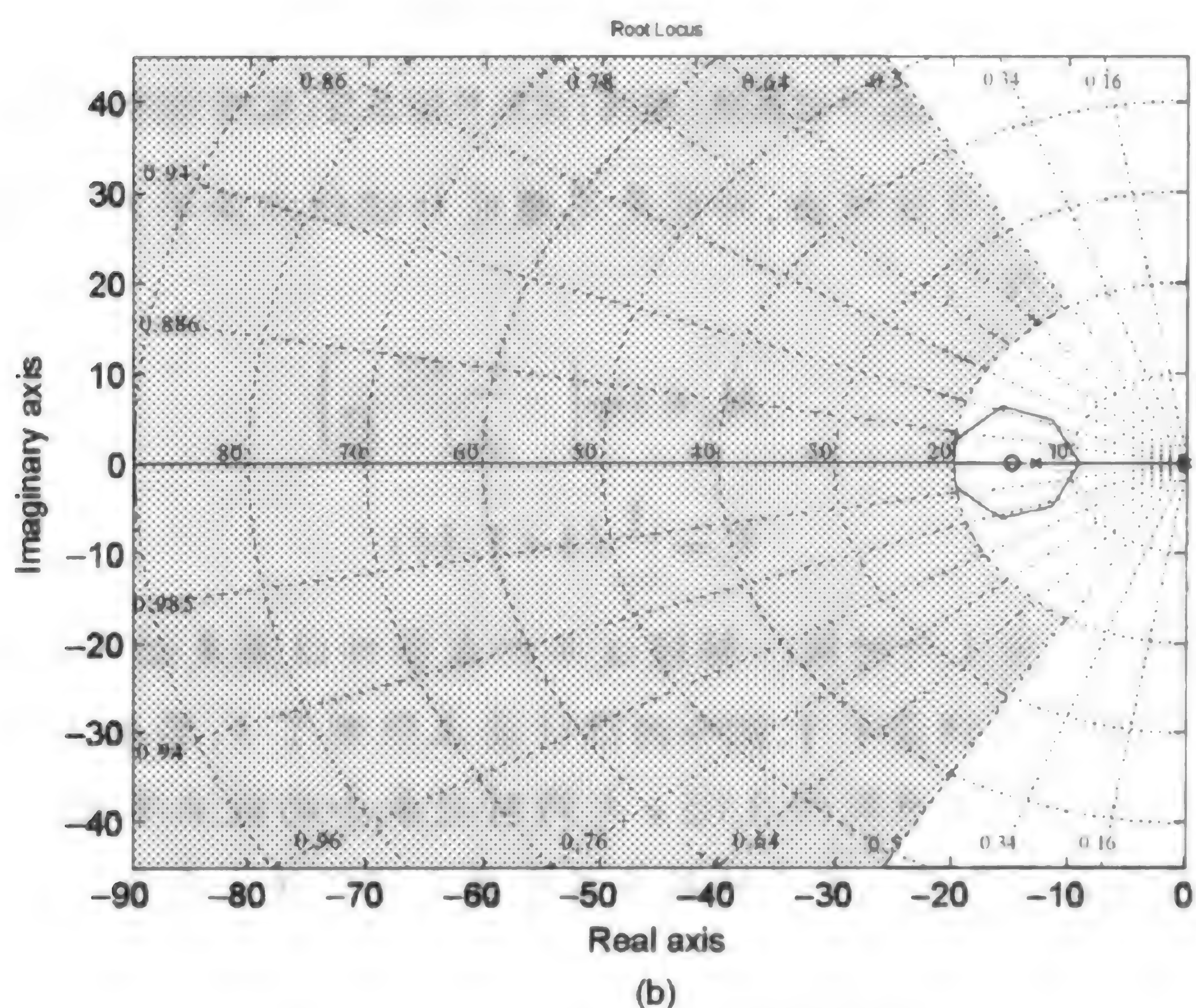
```
theta = linspace( - 2/3 * pi, - 4/3 * pi,15);
X = [20 * cos(theta),200 * cos( - 4/3 * pi),200 * cos( - 2/3 * pi),20 * cos(2/3 * pi)];
Y = [20 * sin(theta),200 * sin( - 4/3 * pi),200 * sin( - 2/3 * pi),20 * sin( - 2/3 * pi)];
hold on;
h = fill(X,Y,'c');
alpha(h,0.2);
sgrid;
rlocus(MotorSS);
axis(100 * [ - 1,0, - 1,1]);
ylabel('Imaginary axis');
xlabel('Real axis');
```

① 参见 D. K. Anand 和 R. B. Zmood, 出处同上, 1995 年。

结果如图 10.24(a)所示。



(a)



(b)

图 10.24　电动机位置系统根轨迹图

由此可见,控制器和控制对象 **MotorSS** 交换了计算次序。理论上,这种计算次序的变化不会影响根轨迹。然而,当超前控制器的根轨迹命令先于对象命令执行时,计算过程将十分缓慢,并且可能会导致某些系统死锁。这是由于控制器零点与对象极点靠得太近的缘故。为简化计算闭环极点的过程,需要对控制器的形式进行转化。如果先计算控制器后计算对象,则这种转化将是奇异的,即级联是不可控的。如果变化一下计算次序,则系统变为不可观测的,也就不影响控制器形式的转化。

类似的超前控制器系统可以用下面的程序产生:

```
theta = linspace( - 2/3* pi, - 4/3* pi,15);
X = [20* cos(theta),200* cos( - 4/3* pi),200* cos( - 2/3* pi),20* cos(2/3* pi)];
Y = [20* sin(theta,200* sin( - 4/3* pi),200* sin( - 2/3* pi),20* sin( - 2/3* pi)];
hold on;
 = fill(X,Y,'c');
alpha(h,0.2);
sgrid;
rlocus(MotorSS* tf([6.667,100],[1,100]));
axis(90* [ - 1,0, - 0.5,0.5]);
ylabel('Imaginary axis');
xlabel('Real axis');
```

程序的执行结果如图 10.24(b)所示。应注意到对于比例控制器而言,在希望的闭环根轨迹区域内没有增益。超前控制器的基本作用是将根轨迹线拉回到满意的极点区域内。

为了找到适当的增益,使用:

```
rlocus(MotorSS * tf([6.667,100],[1 100]))
[k,p] = rlocfind(MotorSS * tf([6.667,100],[1 100]))
```

将十字线放在根轨迹与可接受边缘的交叉点上。函数 rlocus 放在 rlocfind 之前是因为 rlocfind 不画根轨迹,这个过程产生增益 12。这些区域是近似的,因此这是一种通过仿真微调增益的实用方法。从仿真中可以找到可用的最高增益为 13。

在根轨迹设计方法中,目的是将闭环极点放在满意的区域内。针对于状态空间模型开发的线性代数方法,可以将极点直接放在期望的位置就可以解决这类问题。这里介绍的两种极点配置方法采用:

```
place
```

和

```
acker
```

而 LQG 设计方法采用:

```
lqr
lqe
```

和

```
reg
```

函数。

为了满足设计指标选择极点位置为 -30, $-20+30i$ 和 $-20-30i$。这些极点位置是任意选择的,但最好在图 10.24(a)和图 10.24(b)中的阴影区域内。下面的程序用 place 计算增益矩阵 $\boldsymbol{K}$,以使得矩阵$(\boldsymbol{A}-\boldsymbol{BK})$的特征值在期望的位置。矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 如式(10.3)所示。

```
DesiredPoles = [ - 30, - 20 + 30 * i, - 20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles)
```

程序执行后给出 $K=[0.3900, -0.1002, -4.6517]$。假设给定系统的内部状态变量为 x，所需要的反馈为 $u=Kx$。因为上式只能得到输出，所以还需要设计状态观测器。计算控制器和观测器的完整程序如下：

```
DesiredPoles = [ -30, -20 + 30 * i, -20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles);
L = (place(A',C',3 * DesiredPoles))';
ControlSS = reg(MotorSS,K,L);
clSys = feedback(MotorSS,ControlSS, +1);
clSys = 1/dcgain(clSys) * clSys;
step(clSys);
```

作为状态空间控制器，程序的执行结果如图 10.26 所示。程序的前 3 行产生反馈增益矩阵 K。在能观性和能控性之间进行对偶变换，用 place 函数计算反馈观测增益矩阵 L。这个矩阵取决于系统矩阵 A 和 C，以及一些期望的观测器极点位置，在此设为 3 倍于反馈极点的位置。函数 reg 使用对象输出创建了一个观测器，用于估算内部状态并输出正确的值。为了让信号值具有合适的符号应采用正反馈。

前面的例子中，极点位置的选择满足暂态特性要求。实际上，这些选择并不是惟一的。因为在一个区域内，合适的极点位置很多，设计时只是在合适的区域内任意选择一些极点位置而已。线性二次高斯控制器的设计方法遵循类似的步骤，但是为设计者分配极点位置提供了系统的方法。极点的选择可以使下面的积分量最小化①：

$$J=\int[x'(t)Qx(t)+u'(t)Ru(t)]\mathrm{d}t \tag{10.10}$$

其中，$x(t)$ 为 t 时刻系统的内部状态，$u(t)$ 为 t 时刻的输入向量，Q 为半正定矩阵，而 R 为正定矩阵。矩阵 Q 经常选为 $C'C$，以便使上式的第 1 项简化为输出误差的方矩阵。

设计者可以通过修改相应的 Q 和 R 值来调整相应的状态误差输入。特殊情况下，对象只有一个输入，因此 R 为正的标量。当 $Q=C'C$ 时，下面的程序画出最优极点的位置(作为 R 的函数)。此外，极点所处的区域必须满足暂态设计要求。

```
[A,B,C,D] = ssdata(MotorSS);
clPoles = [];
R = logspace( -4,1,60);
for i = 1:length(R)
 [K,S,E] = lqr(A,B,C' * C,R(i));
 clPoles = [clPoles,E];
end;
theta = linspace( -2/3 * pi, -4/3 * pi,15);
X = [20 * cos(theta),200 * cos( -4/3 * pi),200 * cos( -2/3 * pi),20 * cos( -2/3 * pi)];
Y = [20 * sin(theta),200 * sin( -4/3 * pi),200 * sin( -2/3 * pi),20 * sin( -2/3 * pi)];
h = fill(X,Y,'c');
alpha(h,0.2);
hold on;
plot(real(clPoles),imag(clPoles),'x');
axis(40 * [ -1,0, -0.5,0.5]);
```

① 参见 T. Kailith，出处同上，1980 年。

```
sgrid;
ylabel('Imaginary axis');
xlabel('Real axis');
```

程序运行结果的根轨迹如图 10.25 所示，图中用“×”分散地画出了根的位置，而不采用实线。当 R 值很大时，输入所起的作用相对于输出误差所起的作用要大，因而几乎不需要控制即可满足要求。闭环极点与开环极点临近。如果输入所起的作用很小，则最优闭环极点向复平面右半平面移动，需要引入更多的控制作用，且系统响应会更快。因为要同时满足优化条件和暂态特性要求这两个方面，所以必须选择 R 使最优极点位于阴影区域内。

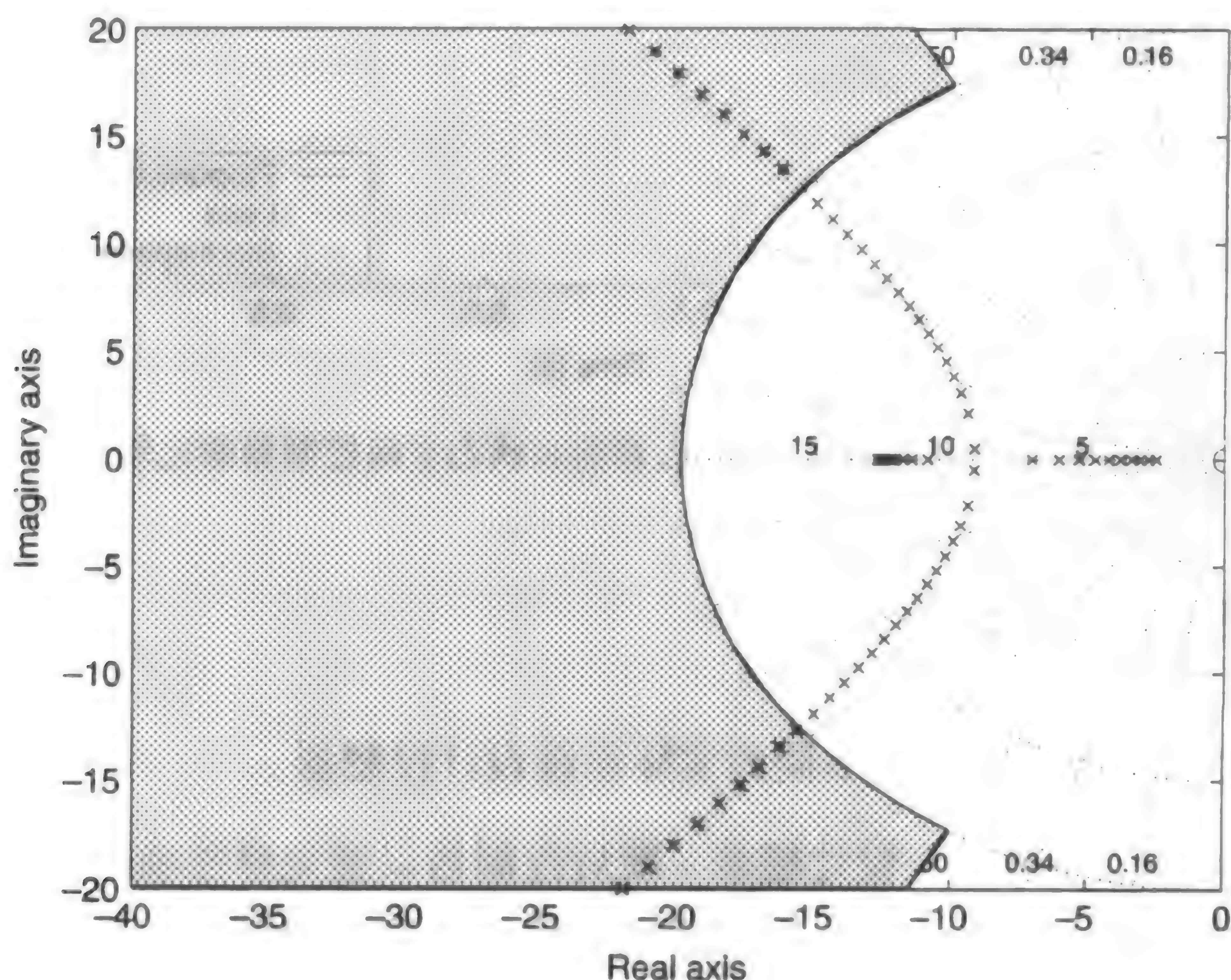


图 10.25 输入代价加权值 R 作为函数时最优控制器的闭环极点

至此，所讨论的三种主要控制器设计方法的阶跃响应曲线由下面的程序产生，并在图 10.26 中进行比较。比例控制器未能满足上升时间设计要求，而采用极点配置的超前控制器和状态空间控制器都满足了设计要求。状态空间控制器的另一个好处是几乎无超调。

```
t = linspace(0,1,200);
yp = step(feedback(1.9416 * MotorSS,1),t);
yl = step(feedback(13.0108 * tf([6.667,100],[1 100]) * MotorSS,1),t);
DesiredPoles = [ - 30, - 20 + 30 * i, - 20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles);
L = (place(A',C',3 * DesiredPoles))';
ControlSS = reg(MotorSS,K,L);
clSys = feedback(MotorSS,ControlSS, + 1);
clSys = clSys/dcgain(clSys);
ys = step(clSys,t);
plot(t,yp,'k-',t,yl,'k--',t,ys,'k-.');
xlabel('Time(s)');
ylabel('Step response');
legend('Proportional','Lead','State-space','Location','SouthEast');
```

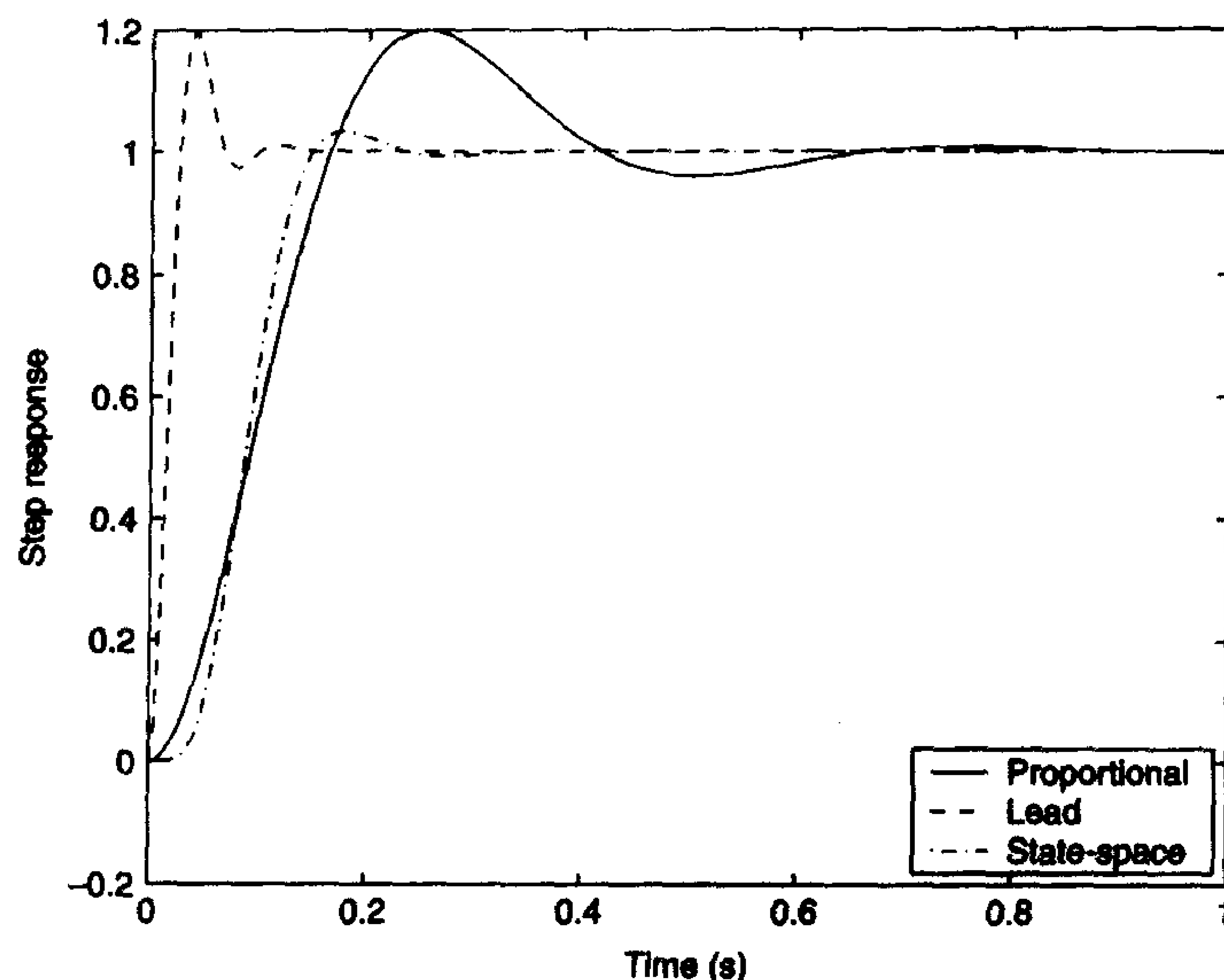


图 10.26 阶跃输入的比例、超前和状态空间控制器的比较

10.5 设计实例

本节将用 MATLAB 工具对 4 个不同的实际系统设计控制器。

1. 带弹性轴的直流电机——设计超前－滞后控制器以避免弹性轴产生振动。
2. 单轴磁悬浮系统——设计 PID 控制器以保持总体的平衡位置。
3. 倒置摆——设计多输入单输出控制器以保持摆的垂直。
4. 磁悬浮飞轮——设计多输入多输出控制器以保持飞轮悬浮。

控制器的设计过程包括 4 个步骤：

1. 确定控制器设计要求
2. 建立对象模型
3. 设计满足要求的控制器
4. 仿真并检验设计的控制器

下面几节中的对象模型由基本原理引伸而来，没有经过实践检验。在控制器的设计中，设计者必须牢记不能过分强调模型的有效性和精确性。上面所描述的后 3 种对象模型是非线性的，但并不严重。在工作点附近每个模型都可被线性化，而且是稳定的。由于有些系统开环不稳定，因此从波特图获得开环数据采用基于频率的设计方法仅适用于带有弹性轴的直流电机系统。

下面几节介绍的控制器包括：

- 超前(滞后)
- 超前－滞后
- PID
- LQG

在上面所述的每一种方法中，根轨迹都是基本的设计方法。实际上，PID 控制器是目前单入单出系统中最常见的控制器类型。

10.5.1 弹性指针的超前－滞后控制

研究硬盘驱动器的读写头。设计目的是让读写头尽可能快地移到期望位置，并为读写操作提供一个稳定的平台。以线圈作为激励，当驱动能力给定时增大移动速度的方法是将固定读写头的旋杆重量减小。旋杆重量的减小使移动加快，但同时也可能使旋杆进入振动状态。

以图 10.27 所示的带有弹性轴的直流电机作为设计模型。用户指定期望的角度 θ_d 和用来测量旋杆偏角 θ 的控制系统，并要求以尽可能快的方式跟踪命令。然而，若弹性指针到达指定位置后长时间振动，则读写头将不能正常工作。

弹性指针的运动方程为①：

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t}+k_b\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}+Ri=v(t)$$

$$J_m\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2}-k_r i(t)=b\left(\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}-\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}\right)+k[\phi(t)-\theta(t)] \tag{10.11}$$

$$J_l\frac{\mathrm{d}^2\phi(t)}{\mathrm{d}t^2}=-b\left(\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}-\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}\right)-k[\phi(t)-\theta(t)]$$

其中，θ 为转子的角度，ϕ 为指针的定位，$J_m=0.03\ \mathrm{kg.m^2}$ 为转动惯量，$J_l=0.015\ \mathrm{kg.m^2}$ 为负荷惯量，$b=0.01$ Nm/rad/s 为弹性轴的阻尼系数，$k=10$ Nm/rad 为弹性轴的恢复常数。电感 L、阻尼 R、反向电动势常数 k_b 及电动机力矩常数 k_r 等各种参数的定义参见 10.2.1 节。

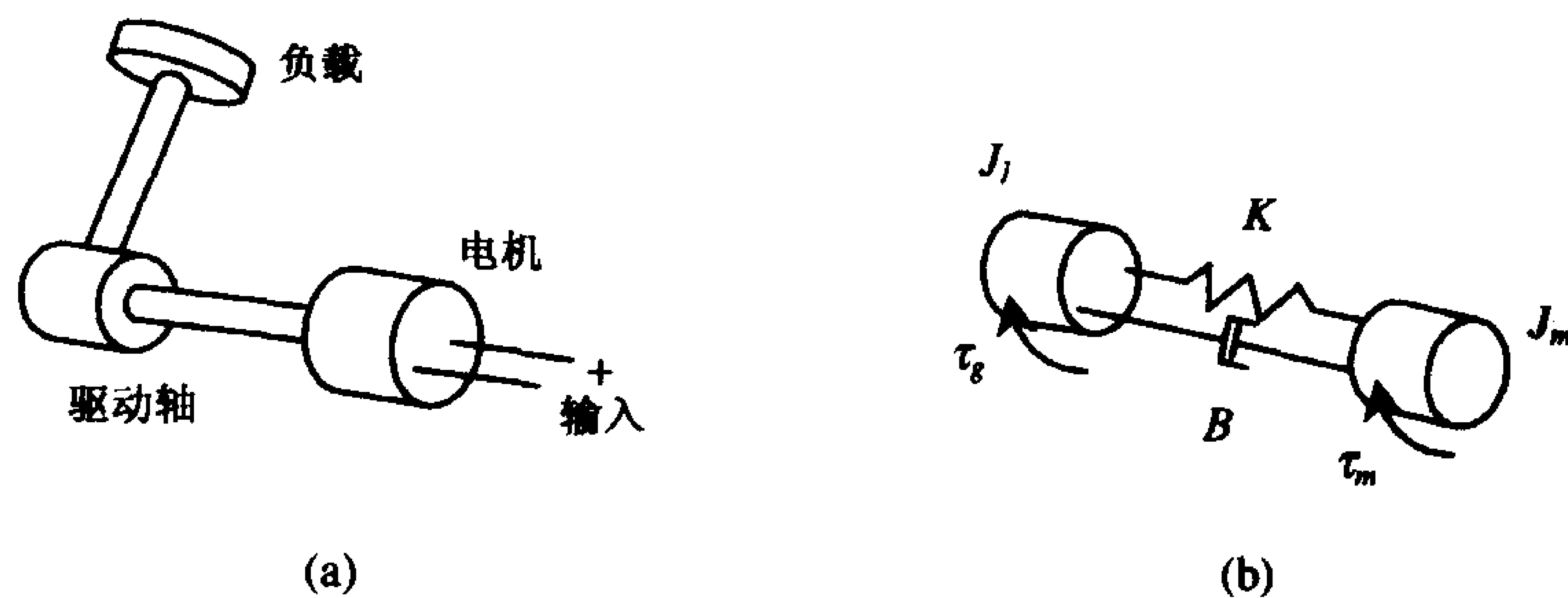


图 10.27 (a) 带弹性驱动轴的指针；(b) 等价模型

通过引入状态变量，可以将耦合的二阶微分方程组转换为线性耦合的一阶方程组：

$$x_1=\theta \qquad x_4=\phi$$

$$x_2=\frac{\mathrm{d}\theta}{\mathrm{d}t} \qquad x_5=\frac{\mathrm{d}\phi}{\mathrm{d}t}$$

$$x_3=i$$

则式(10.11)可转换为：

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}=x_2$$

① 参见 R. C. Dorf 和 R. H. Bishop, *Modern Control Systems*, Addison-Wesley, Reading, MA, 1998 年。

$$\frac{dx_2}{dt} = -\frac{k}{J_m}x_1 - \frac{b}{J_m}x_2 + \frac{k_2}{J_m}x_3 + \frac{k}{J_m}x_4 + \frac{b}{J_m}x_5$$

$$\frac{dx_3}{dt} = -\frac{k_b}{L}x_2 - \frac{R}{L}x_3 + v$$

$$\frac{dx_4}{dt} = x_5$$

$$\frac{dx_5}{dt} = \frac{k}{J_l}x_1 + \frac{b}{J_l}x_2 - \frac{k}{J_l}x_4 - \frac{b}{J_l}x_5$$

下面的函数 **Pointer** 用于计算一阶方程系统矩阵,并返回一个状态空间的系统对象模型。

```
function Plant = Pointer
L = 5e - 3;R = 5;kb = 0.125;
ki = 15;Jm = 3e - 2;
Jl = 0.5 * Jm;k = 10;b = 0.01;
A = [0,1,0,0,0;
     - k/Jm, - b/Jm, ki/Jm, k/Jm, b/Jm;
    0, - kb/L, - R/L,0,0;
    0,0,0,0,1;
    K/Jl, b/Jl,0, - k/Jl, - b/Jl];
B = [0;0;1/L;0;0];
C = [0,0,0,1,0];
D = 0;
Plant = ss(A,B,C,D);
```

系统的输出为指针的角度位置 $\phi(t)$。在 MATLAB 命令窗口键入:

```
pole(Pointer)
```

将显示:

```
1.0e+002 *
  -9.8734
  -0.0248 + 0.3104i
  -0.0248 - 0.3104i
   0.0000
  -0.0871
```

表明弹性指针系统有 5 个极点,其中 3 个极点在实轴上:一个在原点处;另一个由电机引起,在 −1000 rad/s 附近;还有一个由转子动力特性引起,在 − 8.7 rad/s 处。弹性附件增加了一对复数极点 $-2.5 \pm 31.0i$。如前所述,超前控制器可以提高瞬态响应,但阻尼极点的存在将不利于这种方法的使用。

为了说明超前控制在系统中的局限性,有必要画出比例和超前控制系统的根轨迹。将超前零点放在 − 6 正好在第 1 个稳定极点的右侧。超前极点放在 − 50 几乎是零点位置的 10 倍。则传递函数为:

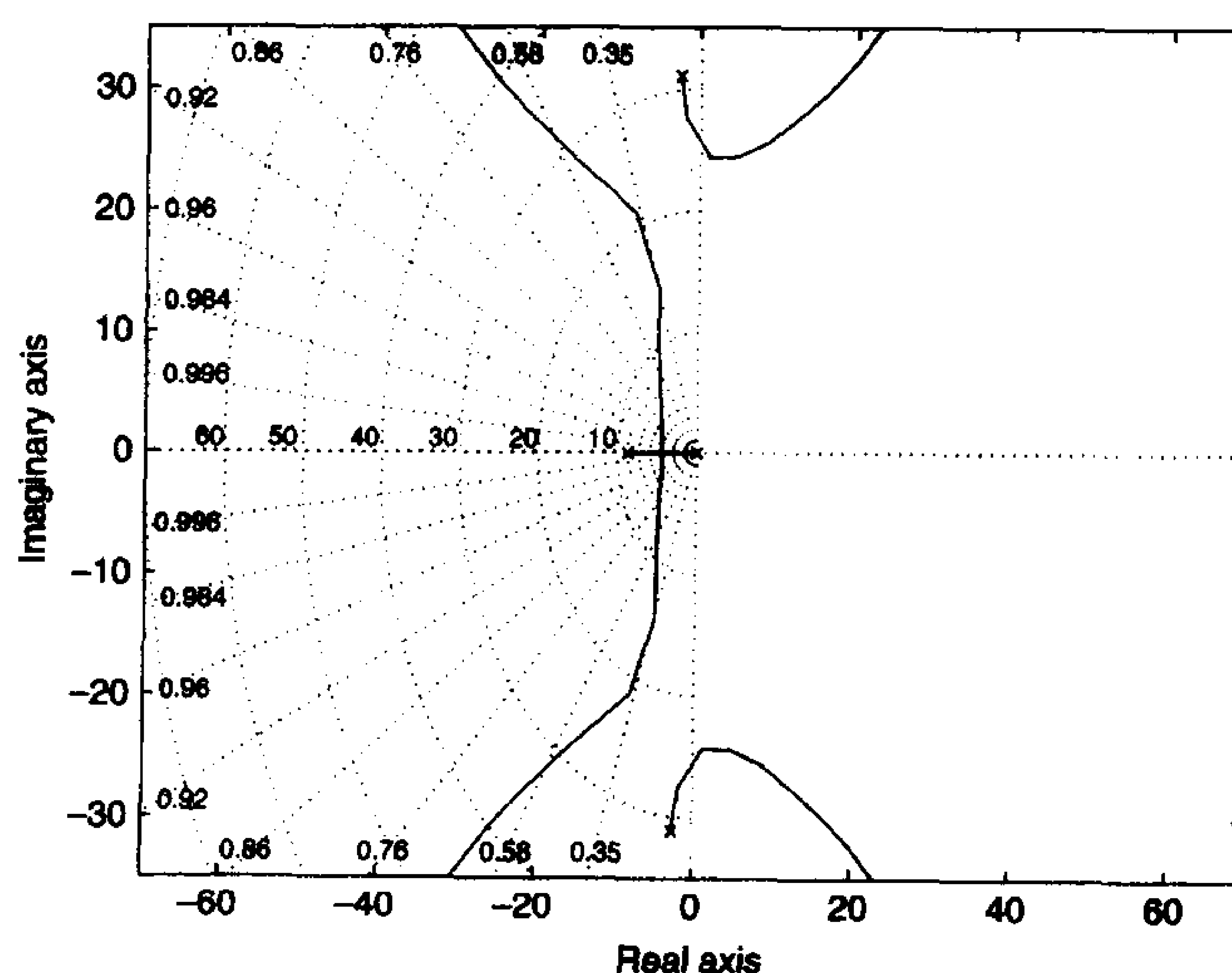
$$H(s) = \frac{s+6}{s+50}$$

程序为:

```
rlocus(Pointer);
```

```
axis(70 * [ - 1,1, - 0.5,0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
figure(2);
rlocus(tf[1,6],[1,50]) * Pointer);
axis(70* [ - 1,1, - 0.5,0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果如图 10.28 所示。对于直流电机系统,两种设计方法所允许的增益不是由开环极点的实部所限制,而是由弹性轴引起的复数极点限制。超前控制器的性能与比例控制器的性能相比略强一些。在 MATLAB 命令窗口中输入:



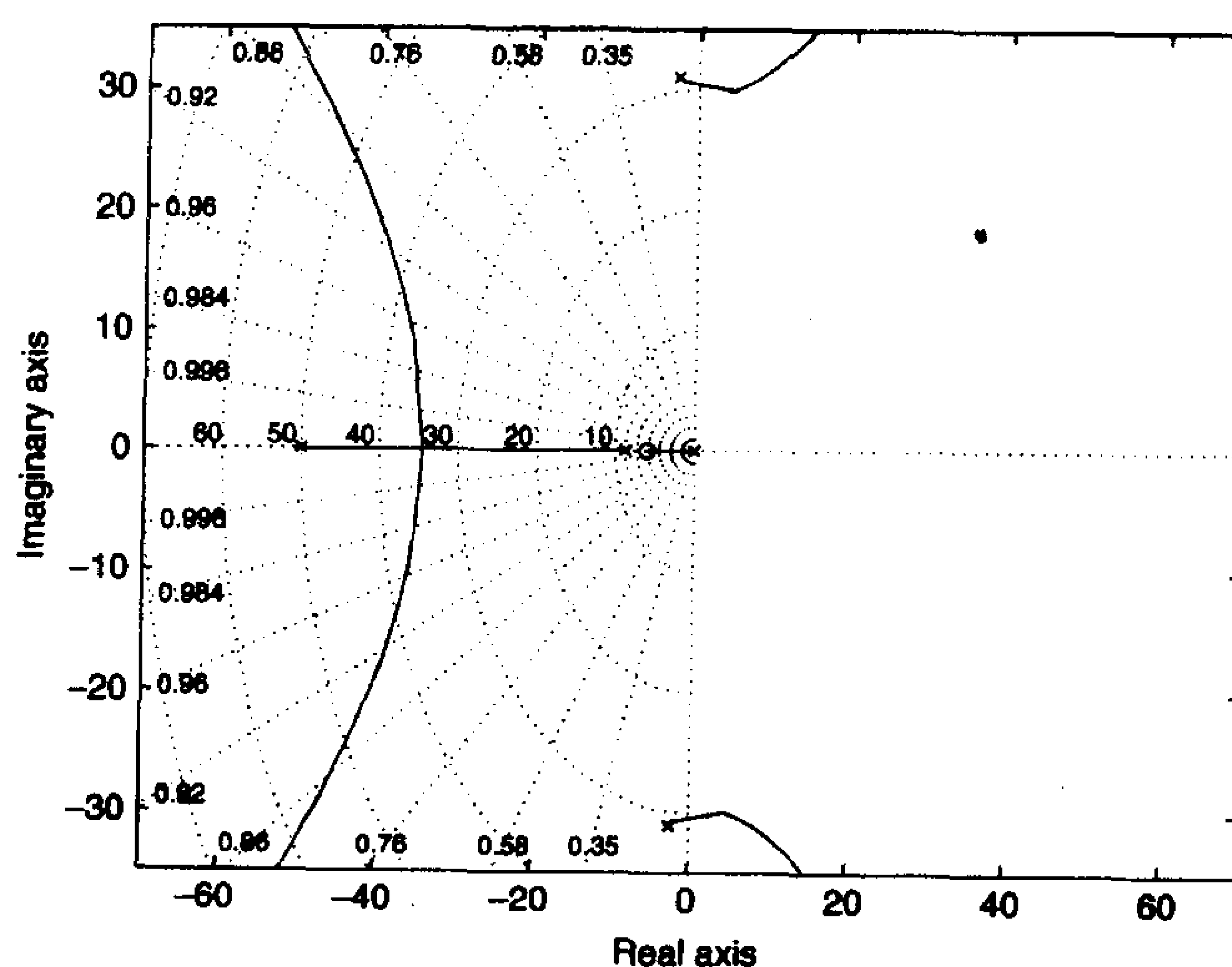


图 10.28 弹性指针的根轨迹:(a) 比例控制;(b) 超前控制

```
rlocus(tf([1 6],[1 50]) * Pointer)
[k,p] = rlocfind(tf([1 6],[1 50]) * Pointer)
```

可以找到超前控制系统和比例控制系统的最佳增益。十字线必须放在由弹性轴引起的复数极点上,因为这是限制增益的主要原因。增益为 3 时,对于每种设计方法系统都是稳定的,并且具有较好的性能,其闭环系统阶跃响应曲线如图 10.30 所示。因为标准控制器无意中会产生自由振荡,所以无阻尼复数极点进一步限制了性能的改善。

不产生自激振荡是进一步提高闭环系统性能的关键问题。回到 10.3.2 节可以看出,如果零点附近正好有一个极点,就不容易产生自激振荡。这里选用超前 - 滞后控制器,其零点选在 $-3 \pm 30i$ 处,在自激振荡位置附近。为保证传递函数的正确性,在 - 60 位置选择两个超前 - 滞后极点,而试图直接消除极点是不现实的。为达到设计要求,必须从一系列实验数据中获得一个良好的模型。

下面的程序绘出了超前 - 滞后控制系统的根轨迹。

```
Notch = zpk([ - 3 + 30i, - 3 - 30i],[ - 60, - 60],1);
rlocus(tf[1 6],[1 50]) * Notch * Pointer);
axis(70 * [ - 1,1, - 0.5,0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果如图 10.29 所示。零点与产生自激的极点非常接近,且系统的根轨迹与没有自激时的根轨迹非常类似。

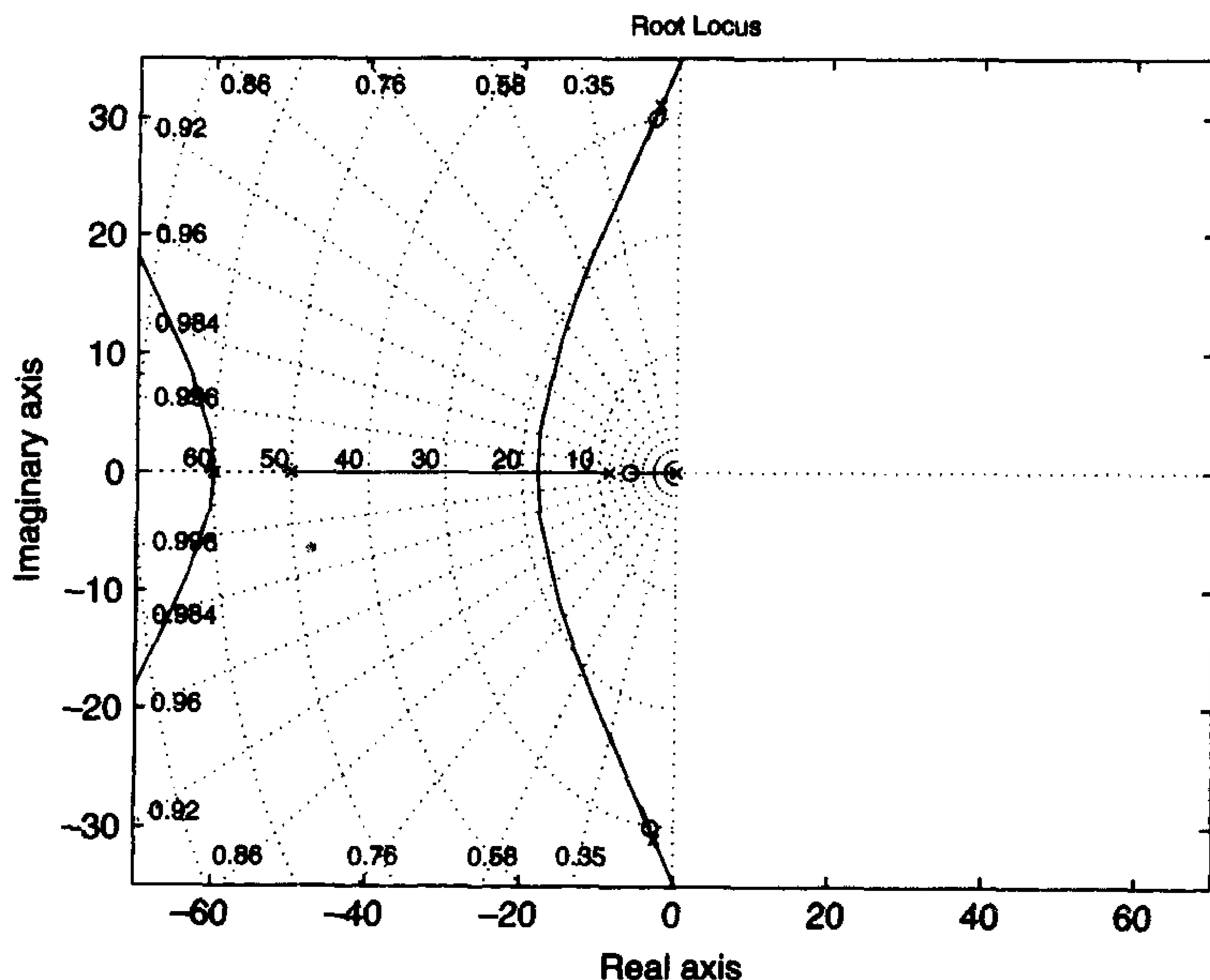


图 10.29 超前 - 滞后超前补偿的弹性指针的根轨迹

用:

```
Notch = zpk([ - 3 + 30i, - 3 - 30i],[ - 60, - 60],1);
rlocus(tf([1,6],[1,50])* Notch* Pointer)
rlocfind(tf([1,6],[1,50])* Notch* Pointer)
```

可找到产生一个稳定闭环系统的增益，值为 40.0。十字线放在根轨迹与虚轴的交叉点上，将其放在以负实轴为中心的 120°楔形内，将得到很好的暂态特性。

下面的程序用于计算前面所讲述的 3 种控制方案的阶跃响应。

```
Lead = tf([1,6],[1,50]);
Notch = zpk([-3+30i, -3-30i],[-60, -60],1);
t = linspace(0,3,200);
yp = step(feedback(3.0* Pointer,1),t);
yl = step(feedback(3.0* Lead* Pointer,1),t);
yn = step(feedback(40.0* Notch* Lead* Pointer,1),t);
plot(t,yp,'k--',t,yl,'k-.',t,yn,'k-');
legend('Proportional','Lead','Notch');
xlabel('Time');
ylabel('Step response');
```

程序执行结果如图 10.30 所示。超前 - 滞后控制器具有同比例控制器相同的上升时间，但却没有引起轴的自激振荡。

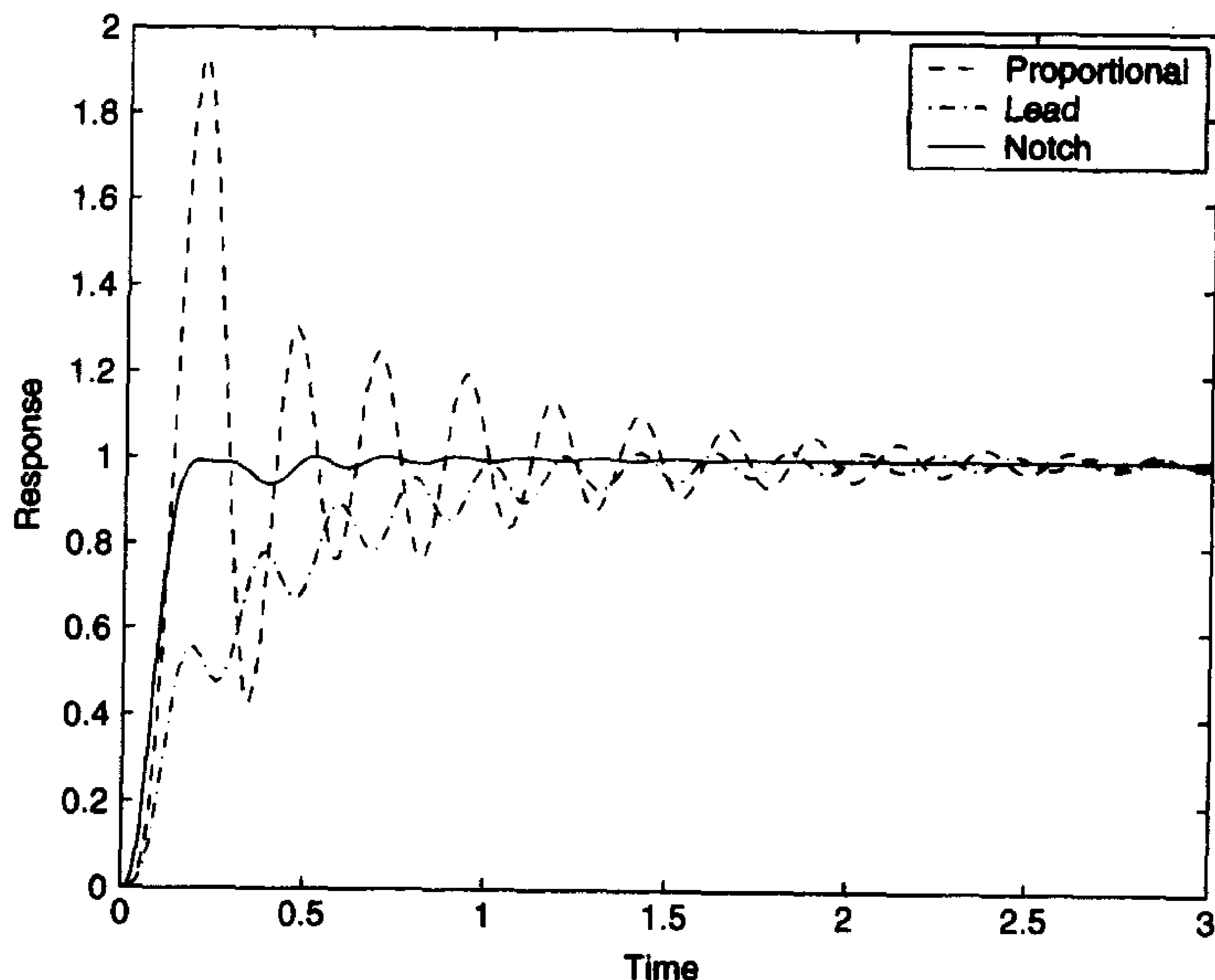


图 10.30 比例、超前、超前 - 滞后控制器阶跃响应比较

前面的设计只是构成了模拟的控制器，而最终的控制器应该是嵌入式的、廉价的小型计算机。嵌入式控制器用光学编码器周期性地测量指针的位置，并与期望位置相比较。通过计算，由数/模转换器或脉冲宽度调制器可设置有效的放大器电压。在下面的例子中，微机读取编码器的编码结果，并以每秒一百次的速度更新电压放大器的输出。由于采样时间间隔很小，因此应尽量减小采样间隔内的计算时间。下面的程序设计了对象控制器的离散化模型，并将数字化设计与前面的连续超前 - 滞后设计进行了比较。数字超前 - 滞后滤波器的零点设置在 $0.92 \pm 0.3i$ 处，与对象离散化模型的极点相接近，极点设置为 0.6。数字化传递函数的实现不是绝对固定的。数字超前控制器的零点设置为 0.95，刚好在离散化对象第一个稳定极点 0.5 的右侧。利用：

```
Ts = 0.01;
DNotch = zpk([0.92 + 0.3i,0.92 - 0.3i],[0.6 0.6],1,Ts);
DLead = tf([1, - 0.95],[1, - 0.5],Ts);
rlocus(DNotch* DLead* c2d(Pointer,Ts))
rlocfind(DNotch* DLead* c2d(Pointer,Ts))
```

将十字线放在远离单位圆的根轨迹上,可得出增益 15。

```
Ts = 0.01;
DNotch = zpk([0.92 + 0.3i,0.92 - 0.3i],[0.6 0.6],1,Ts);
DLead = tf([1, - 0.95],[1, - 0.5],Ts);
rlocus(DNotch* DLead* c2d(Pointer,TS))
axis(1.2* [ - 1,1, - 1,1]);
zgrid;
figure(2);
[yd,t] = step(feedback(15 * DNotch * DLead * c2d(Pointer,Ts),1));
Lead = tf([1,6],[1,50]);
Notch = zpk([ - 3 + 30i,-3-30i],[ - 60, - 60],1);
yn = step(feedback(40.0 * Notch * Lead * Pointer,1),t);
plot(t,yd,'k-',t,yn,'k:');
legend('Discrete control','Continuous control','Location','SouthEast');
xlabel('Time(s)');
ylabel('Response');
axis(1.2* [0,1,0,1]);
```

程序执行结果如图 10.31 所示。

10.5.2　磁悬浮系统的 PID 控制

研究图 10.32 所示的磁悬浮系统①。其目的是让悬浮的球在受到外界干扰时仍然在期望的高度。球的高度为 $h(t)$,线圈电流为 $i(t)$。磁悬浮运动方程为:

$$m\frac{\mathrm{d}^2h(t)}{\mathrm{d}t^2} = mg - k\left(\frac{i(t)}{h(t)}\right)^2 \tag{10.12}$$

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t} = v(t) - Ri(t)$$

其中,m 为球的质量,g 为重力加速度,L 为线圈电感,R 为线圈电阻,k 为磁场和球之间的耦合系数。系统输入为线圈电压 $v(t)$,输出为球的高度 $h(t)$。方程为非线性的。

磁场越强,球与电磁体的距离越近,这将破坏系统的稳定性。理想情况下,球与磁铁间应保持足够的距离以使磁力抵消地球的引力。如果球下降得太多,则磁场的作用会变弱而使球完全掉下去。如果球离磁铁太近,则磁场作用又会太强而将球拉向磁铁。因此,首先要计算出地球引力和磁场引力相等的点,这个点称为平衡点。给定期望位置 h_0,可通过令加速度等于零的方法,求出维持该位置的期望电流。即:

$$i_0^2 = \frac{mg}{k}h_0^2$$

注意,模型可以在平衡点附近线性化,对式(10.12)线性化可产生一组线性方程。

① 参见 B. Friedland, *Advanced Control System Design*, Prentice Hall, Englewood Cliffs, NJ, 1996 年。

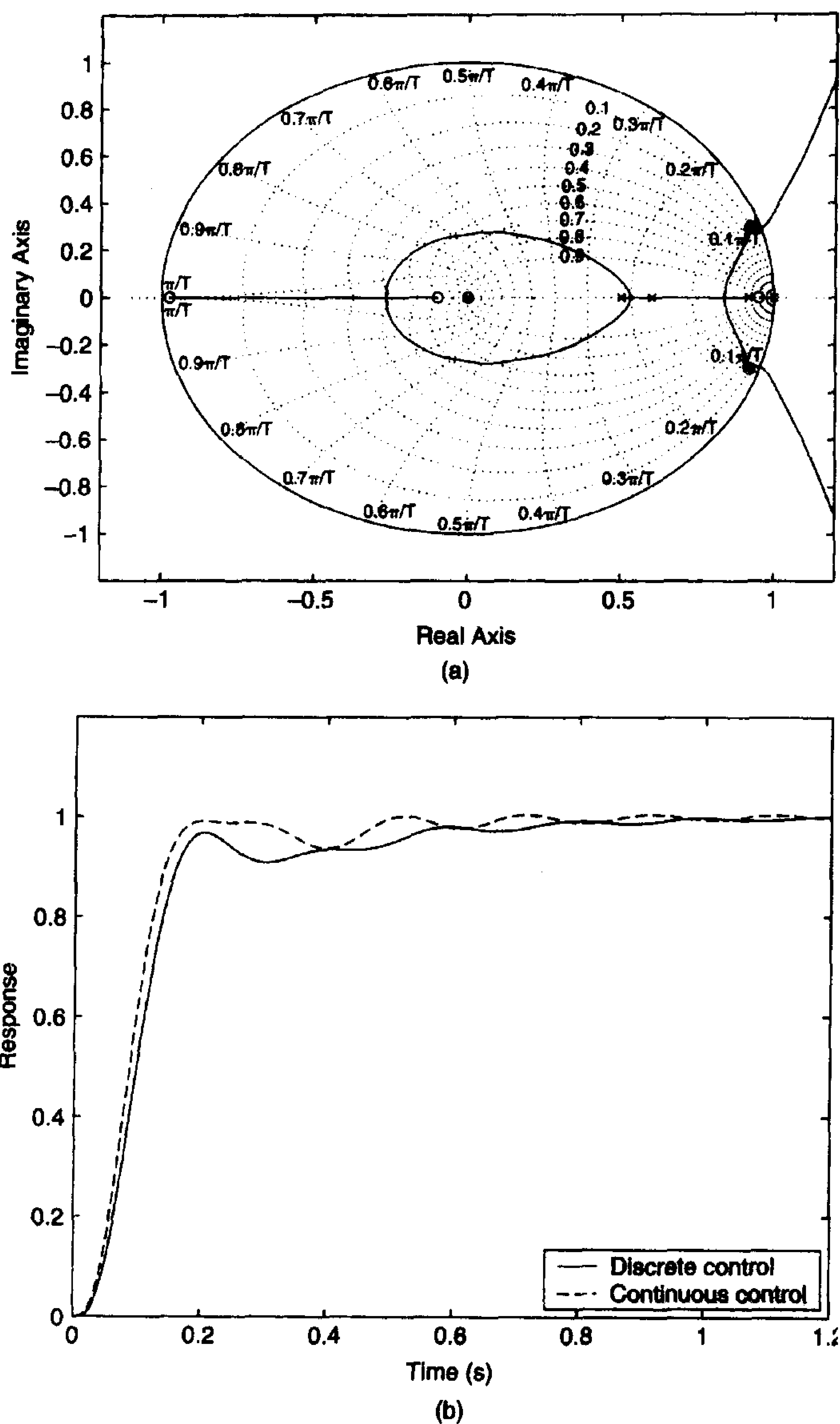


图 10.31 弹性指示器系统的数字实现:(a) 根轨迹;(b) 阶跃响应

引入状态变量:

$$x_1 = h \quad x_2 = \frac{\mathrm{d}h}{\mathrm{d}t} \quad x_3 = i$$

则式(10.12)变成:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= x_2 \\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= g - \frac{k}{m}\left(\frac{x_3}{x_1}\right)^2 \\
\frac{\mathrm{d}x_3}{\mathrm{d}t} &= \frac{v}{L} - \frac{R}{L}x_3
\end{aligned} \tag{10.13}$$

式(10.13)可以在工作点 $x_3 = i_0, x_1 = h_0$ 处通过泰勒级数展开进行线性化。线性化将导致 dx_2/dt 的更改,线性化结果为:

$$\begin{bmatrix} dx_1/dt \\ dx_2/dt \\ dx_3/dt \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m}\frac{i_0^2}{h_0^3} & 0 & \frac{-2k}{m}\frac{i_0}{h_0^2} \\ 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix}$$

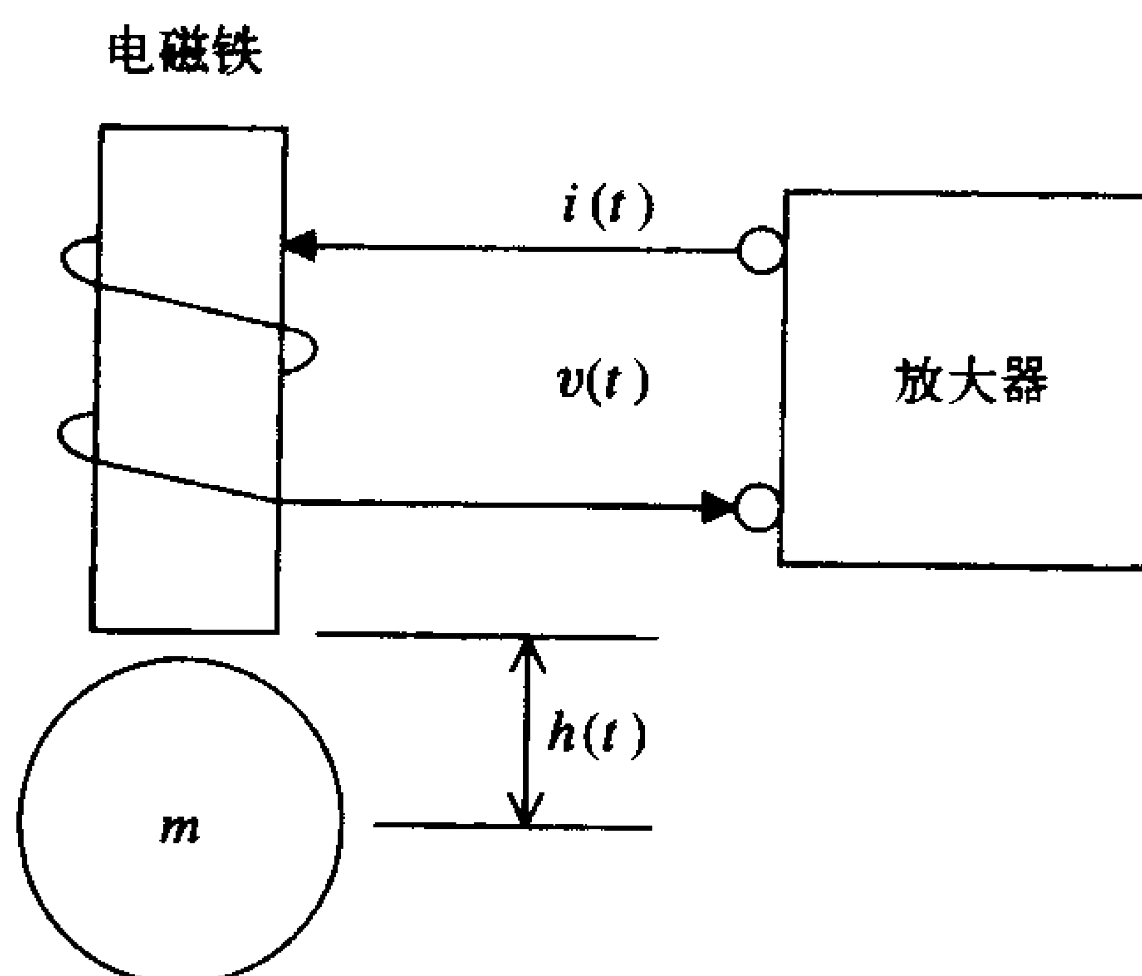


图 10.32　磁悬浮系统

假设球的质量为 100 gm,线圈电阻为 5 Ω,线圈电感为 40 mH,耦合系数为 0.01 Nm^2/A,期望高度为 2 cm。首先,建立函数 **MagLev** 描述系统状态空间模型。

```
function Plant = MagLev
m = 0.1;g = 9.82;R = 5;
L = 0.04;k = 0.01;h0 = 0.02;
i0 = h0 * sqrt(m * g/k);
A = [0,1,0;2 * k * i0^2/(m * h0^3),0, - 2 * k * i0/(m * h0^2);0,0, - R/L];
B = [0;0;1/L];
C = [1,0,0];
D = 0;
Plant = ss(A,B,C,D)
```

在 MATLAB 命令窗口键入:

```
MagPoles = pole(MagLev)
```

得到:

```
MagPoles =
   31.3369
  -31.3369
  -125.0000
```

因此,线性化后悬浮系统的极点为 ±31.3,而放大器的极点为 -125。故需要加入一个 PD 控制器来稳定系统。理论上,PD 控制器的传递函数由下式给出:

$$C_0(s) = k_p + sk_d$$

其中,k_p 为比例增益,k_d 为微分增益。控制器的输出是对输入的微分,由于高频噪声的影响,这在实际中是很难实现的。一般来说,微分作用可以近似并可通过滤波的方法消除噪声的影

响。其结果为：

$$C_1(s)=k_p+k_d\frac{s}{\tau_f s+1}=k_p\frac{(\tau_f+k_d/k_p)s+1}{\tau_f s+1}$$

传递函数 $C_1(s)$ 等价于一个超前控制器，其时间常数为 τ_f+k_d/k_p 的零点和时间常数为 τ_f 的极点(滤波器)。之所以选用超前控制器是因为零点总是比极点慢。选择控制器零点为 -20，刚好在磁悬浮系统第一个稳定极点的右侧，滤波器极点为 -50，则 $\tau_f=20$ ms。为保证系统在正反馈时稳定，在控制器中要改变符号，传递函数为：

$$C_1(s)=-\frac{s+20}{s+50}$$

程序为：

```
PD = tf( - 1 * [1,20],[1,50]);
rlocus(PD * MagLev);
sgrid
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果的根轨迹如图 10.33 所示。

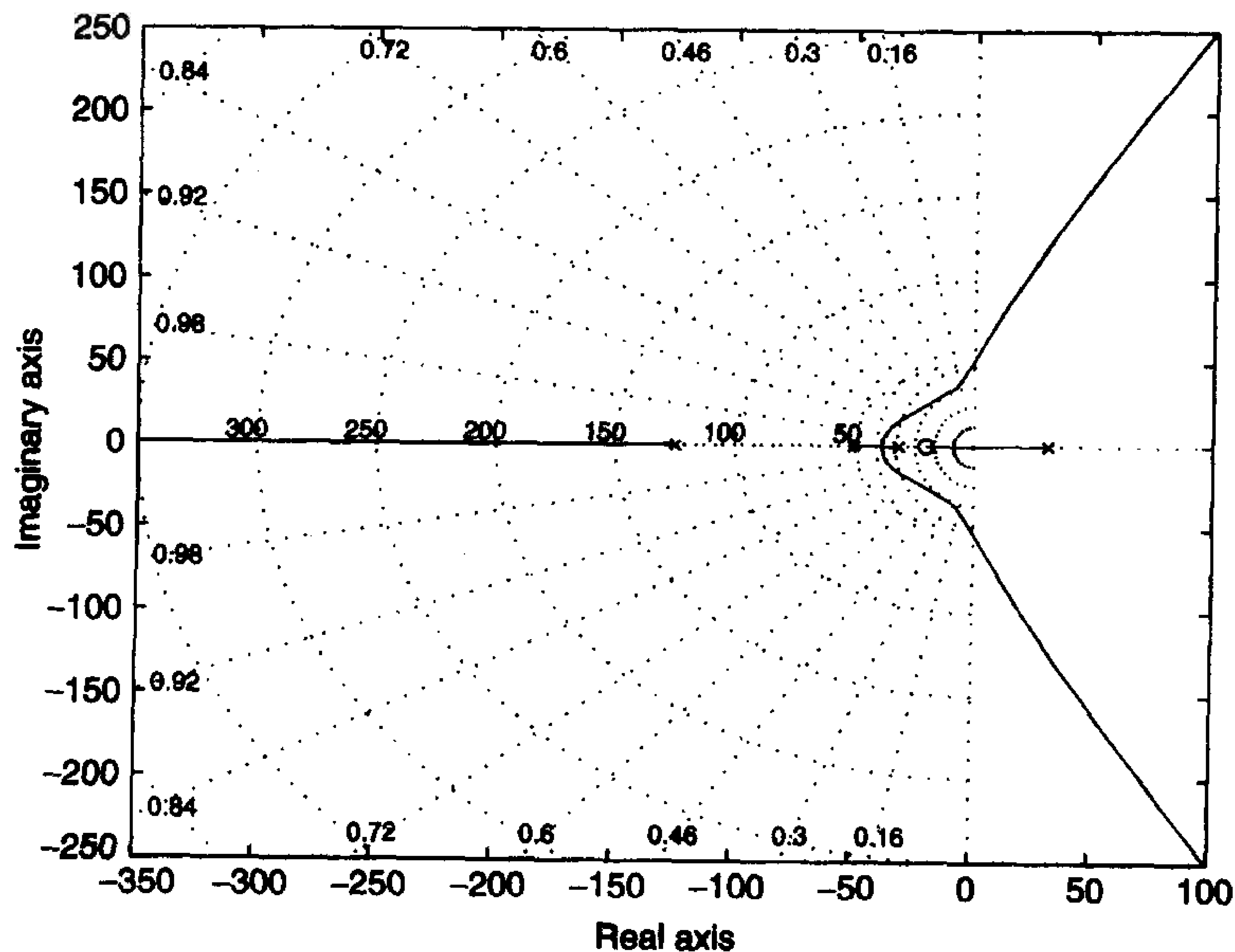


图 10.33　PD 串联控制器的磁悬浮系统根轨迹

从图中可以看出，在某些低增益情况下，不稳定极点移到了左半平面；而在高增益时，复数共轭极点则进入了右半平面。在 MATLAB 命令窗口键入：

```
rlocus(tf( - 1 * [1,20],[1,50]) * MagLev)
rlocfind(tf( - 1 * [1,20],[1,50]) * MagLev)
```

并将十字线放在实轴根轨迹上，介于不稳定极点与控制器零点之间，则可由图中得到一个稳定的增益值。在虚轴和控制器零点之间近于一半的位置，产生增益 150。因此，$k_p=-60$，$k_d=-1.8$，且滤波器时间常数为 20 ms，即相当于 1/50。

在测定 $v_0 = Ri_0$ 时,最可能引起测量误差和模型化误差,从而使球的位置 $h(t)$产生稳态误差。PD 影响系统的不稳定性和暂态特性,而 PI 则经常用于改善系统的稳态特性。研究 PI 控制器:

$$C_2(s) = k_p + \frac{k_i}{s} = \frac{k_p}{k_i}\frac{s + k_i/k_p}{s}$$

它带有参数 k_p 和 k_i。PI 控制器有一个极点在原点处,一个零点在 $-k_i/k_p$ 处。相对于系统其他零极点而言,如果 PI 控制器零点与极点相距很近,则当 PI 控制器与 PD 控制器串联形成 PID 控制器时,对于闭环暂态特性的影响是可以忽略的。因此,反馈增益 150 仍然可以使用。PI 控制器对于稳态误差的影响很大,在该控制系统中选择 $k_p = k_i = 1$。下面的程序仿真了线性磁悬浮模型闭环系统的脉冲响应:

```
PD=tf(-1*[1,20],[1,50]);
PI=tf([1,1],[1,0]);
[y,t]=impulse(feedback(150*PI*PD*MagLev,1));
plot(t,y,'k-');
grid;
xlabel('Time');
ylabel('Impulse response');
```

程序执行结果如图 10.34 所示。结果返回闭环系统的线性化模型,但更为重要的是确定回路中非线性模型系统的稳定性。

非线性系统可以用 SIMULINK 进行仿真。为此,需要用到一个称为 *S-Function* 的模块,该模块是由用户定义的函数。这里将该用户定义的函数称为 **MagModel**。SIMULINK 的 *S-Function* 模块指定需要对 **MagModel** 函数按以下顺序传递 4 个参数:时间 t、状态变量 x、输入量 u 和一个整数 *flag*。离散和连续状态的混合模型可以用 *S-Function* 建立。仅研究连续非线性化模型部分,SIMULINK 查询用户函数以确定非线性模型,*flag* 决定查询的目的。当 *flag* = 1 时,函数返回由式(10.11)确定的 x 关于 t 的导数、状态 x 和输入量 u;当 *flag* = 3 时,函数返回输出量;当 *flag* = 0 时,函数返回向量 *sys*,其元素依次为:连续状态数、离散状态数、输出量数目、输入量数目、根的数目和一个最终标识,当系统有直接反馈时该标识置 1。在磁悬浮问题中,*sys* = [3 0 1 1 0 0],说明有三个连续状态,没有离散状态,一个输入,一个输出,无根且没有直接反馈。当 *flag* = 0 时,也返回连续状态的初始条件,即系统第一次运行时的状态。系统的平衡位置为 $x(0) = [h(0)\ 0\ i(0)]'$。启动系统时不在其平衡位置,而是赋予 $h(0)$一个比平衡值大 10% 的初始值。在这样的初始条件下,控制器会开始工作,否则球会脱离磁体。因为控制器是基于系统的线性模型设计的,所以平衡位置的偏差一定要小。

```
function[sys,x0]=MagModel(t,x,u,flag)
m=0.1;g=9.82;R=5;L=0.040;
k=0.01;h0=0.02;
i0=h0*sqrt(m*g/k);
switch flag
 case 1
  xdot=zeros(3,1);
  xdot(1)=x(2);
  xdot(2)=m*g-k*x(3)^2/x(1)^2;
  xdot(3)=-R/L*x(3)+1/L*u(1);
  sys=xdot;
case 3
```

```
    sys = x(1);
case 0
    sys = [3 0 1 1 0 0];
    x0 = [h0 + 0.1 * h0;0;i0];
otherwise
    sys = [];
end
```

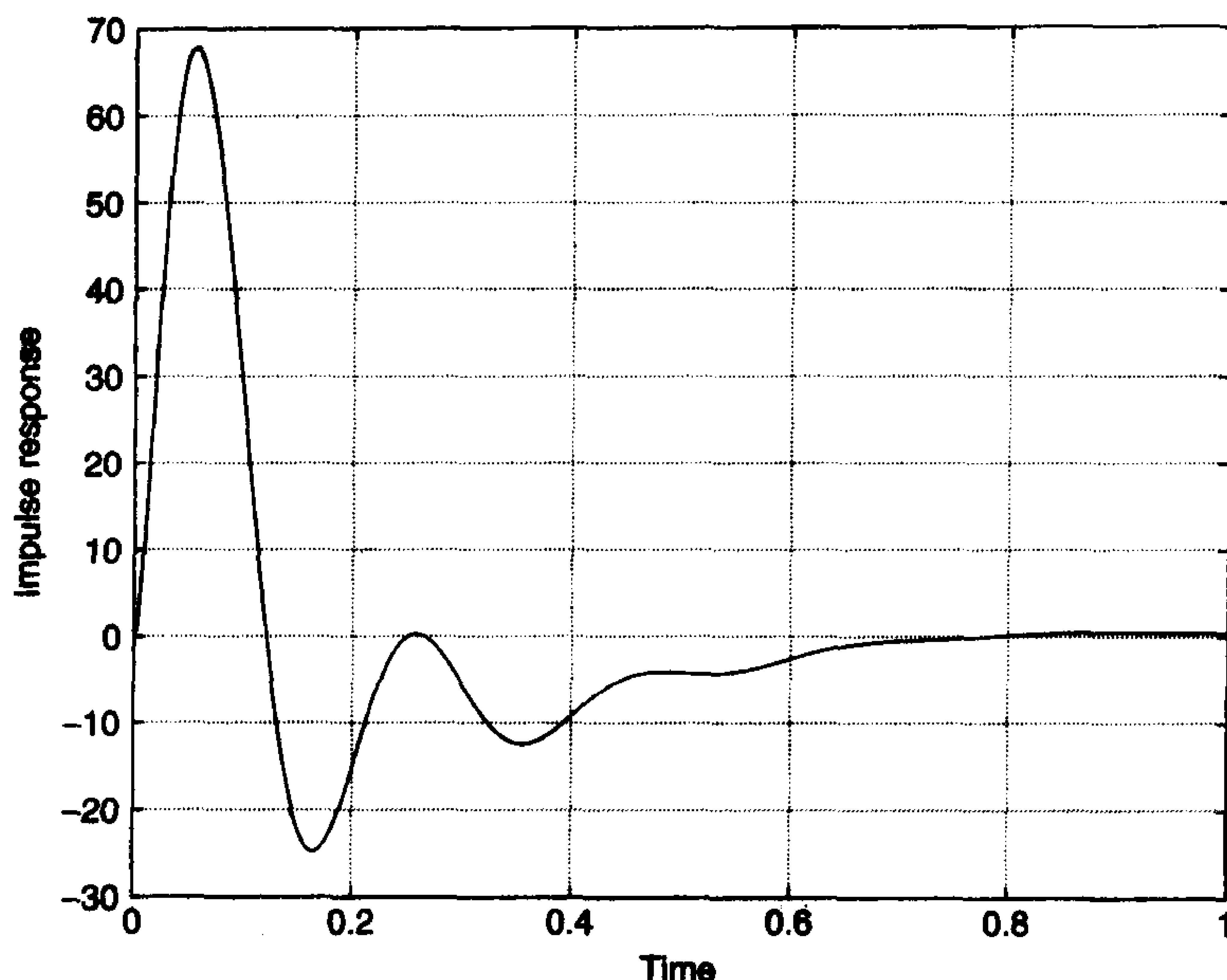


图 10.34 磁悬浮系统线性模型的脉冲响应曲线

控制器由变量名而不是由数值确定，因此，这些变量在 SIMULINK 运行前必须要在 MATLAB 命令窗口中定义。运行下面的程序可生成这些变量，并运行 SIMULINK：

```
PD = tf( - 1 * [1,20],[1,50]);
PI = tf([1,1],[1,0]);
v0 = 0.991;h0 = 0.02;
[num,den] = tfdata(150 * PD * PI,'v');
simulink;
```

$v0$ 和 $h0$ 的值由给定值算出。

用 MATLAB 建立如图 10.35 所示的方框图。从 *Math* 组件库选用 *Sum* 和 *Gain* 模块。从 *User Defined Functions* 组件库选用 *S-Function* 模块，双击模型窗口的 *S-Function* 模块并用 **MagModel** 作为 *S-Function* 模块的名称。

从 *Continuous* 组件库选用 *Transfer Fnc* 模块，双击此模块，并分别输入分子和分母的变量名，本例中为 *num* 和 *den*。在进入 SIMULINK 前，在程序的第 3 行对这两个数进行了赋值。

从 *Sources* 组件库中选用 *Constant* 模块，并输入电压偏差和高度偏差，这些值用来描述误差。如果变量名已经或即将在 MATLAB 命令窗口中赋值，可通过双击模块输入其数值或输入其变量名。因为 $v0$ 和 $h0$ 在前面运行的程序中已经定义，所以选用后一种方法。

最后，从 *Sinks* 组件库中选用 *To Workspace*，记录作为时间函数的承载高度（*Height*）和磁铁线圈电压（*Vapp*）的幅值。时间值来自 *Sources* 组件库的 *Clock*，并且存储在 *Time* 数组中。三个

模块周围的阴影是通过 *Format* 下拉菜单的 *Show Drop Shadow* 命令得到的。在每一个仿真时刻，*Vapp* 和 *Height* 的值保存在工作空间里，并可以通过函数 Plot 来显示。图 10.36 显示了 *Vapp* 和 *Height* 随时间变化的曲线，是通过在 MATLAB 命令窗口中运行下列程序得到的。*Height* 的值进行了 50 倍放大。

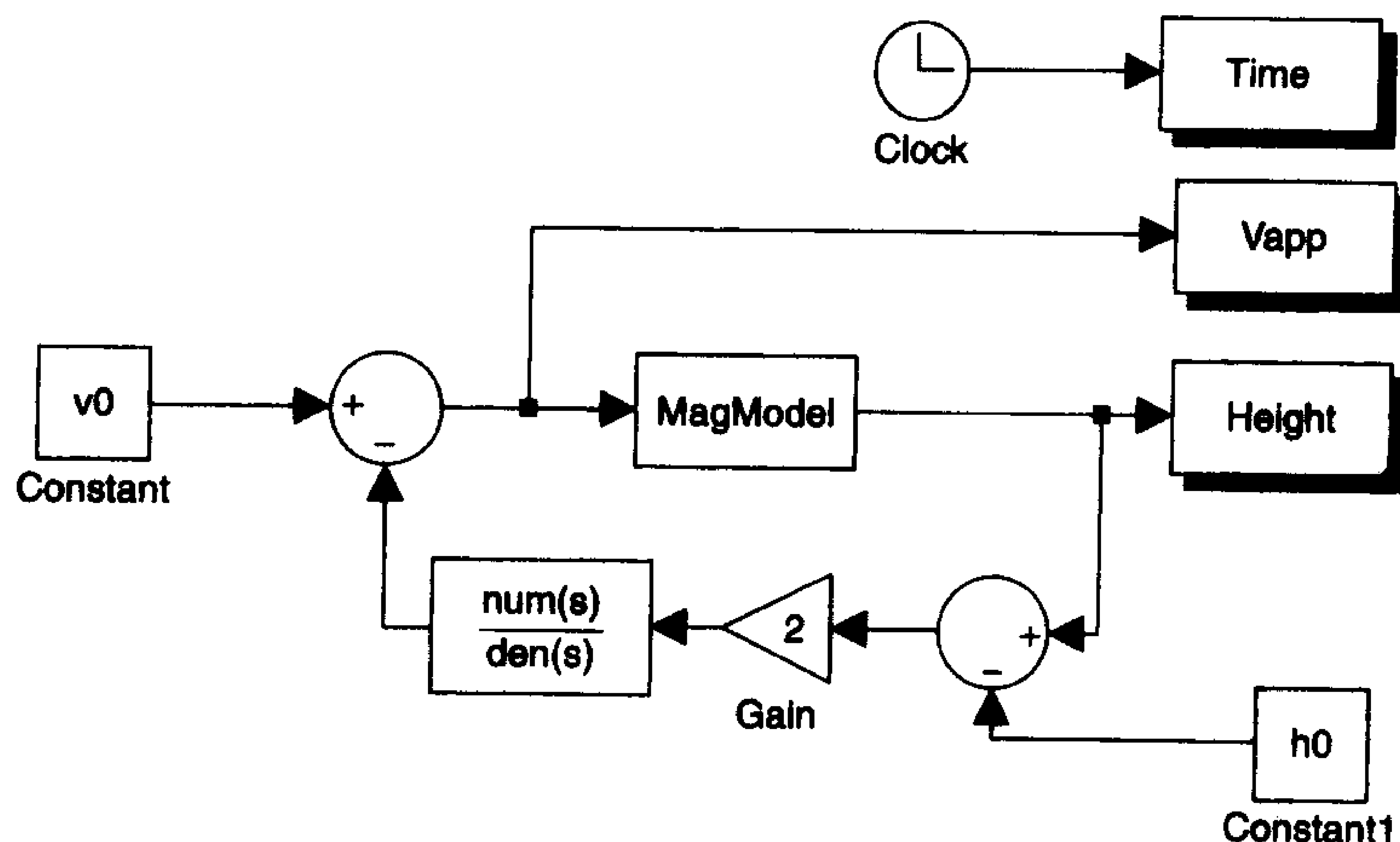


图 10.35 非线性磁悬浮系统的 SIMULINK 方框图

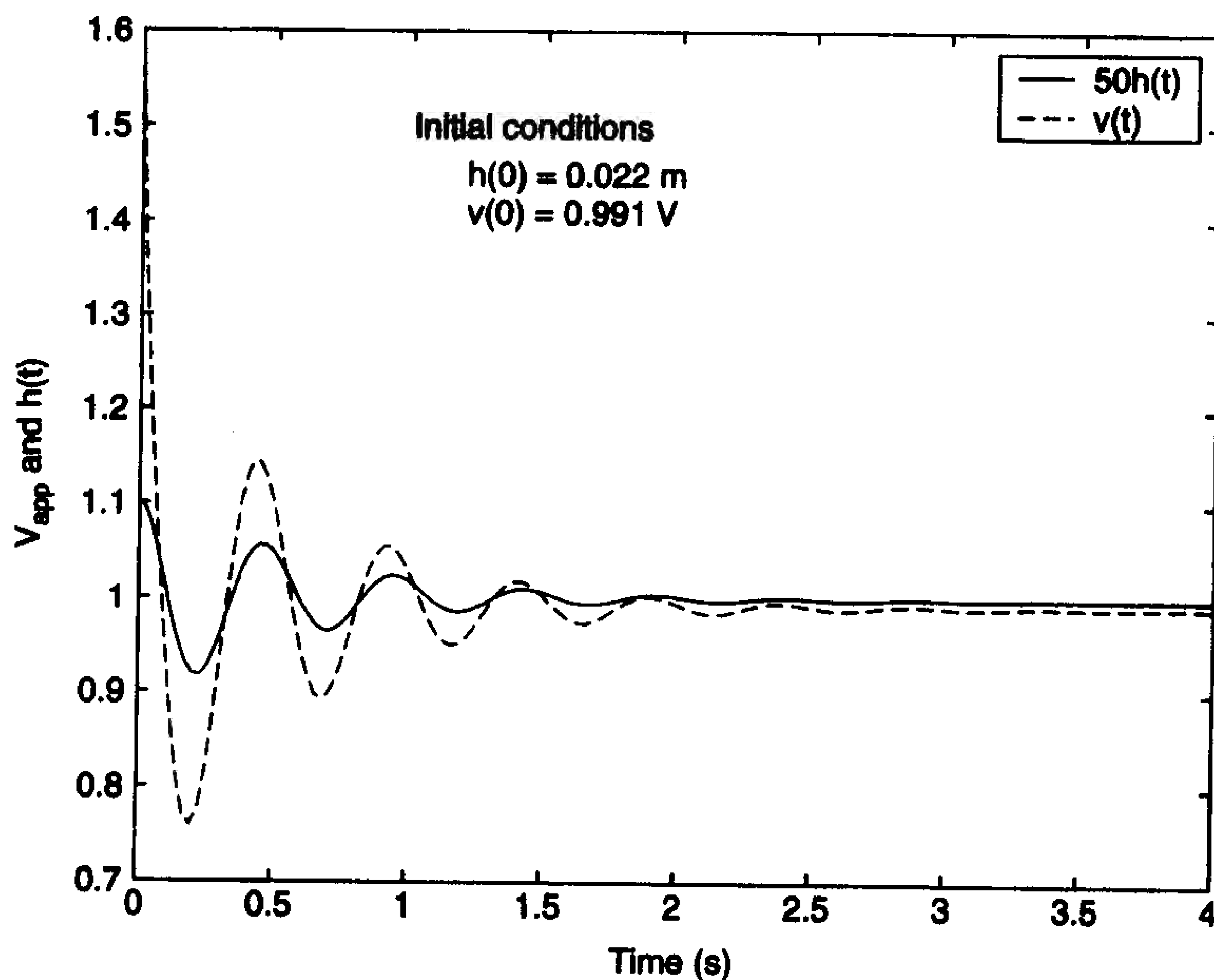


图 10.36 带有初始条件和模型误差的非线性磁悬浮系统在 PID 控制下的响应曲线

```
plot(Time,50 * Height,'k-',Time,Vapp,'k-.')
legend('50h(t)','v(t)')
text(1,1.5,'Initial conditions')
text(1.2,1.45,'h(0) = 0.022 m')
text(1.2,1.4,'v(0) = 0.991 V')
xlabel('Time(s)')
ylabel('V_{app} and h(t)')
axis([0 4.7 1.6])
```

10.5.3　倒置摆的超前控制

用根轨迹方法可为倒置摆设计超前控制器。研究图10.37所示的装在圆盘上的倒置摆。控制系统的目标是通过控制圆盘的位置保持摆的垂直。圆盘的角度 Ψ 和摆的角度 θ 都是可以测量的,其运动方程为:

$$ml^2\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}+mrl\cos(\theta)\frac{\mathrm{d}^2\Psi}{\mathrm{d}t^2}=mgl\sin(\theta)+b_1\frac{\mathrm{d}\theta}{\mathrm{d}t}$$

$$mrl\cos(\theta)\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}+(J+mr^2)\frac{\mathrm{d}^2\Psi}{\mathrm{d}t^2}=mrl\sin(\theta)\left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2+b_2\frac{\mathrm{d}\Psi}{\mathrm{d}t}+\tau_m \qquad (10.14)$$

其中,m 为摆锤的质量,l 是摆的长度,r 是圆盘的半径(也可以是摆锤连接部分的半径),d 是圆盘的厚度,$J=\rho\pi dr^4/4$ 为圆盘的惯量,b_1 是摆旋转的摩擦系数,b_2 是圆盘旋转的摩擦系数,τ_m 是与圆盘底部相连的电动机的转力矩。与磁性轴承类似,当 θ 和 $\mathrm{d}\theta/\mathrm{d}t$ 很小时在工作点附近运动方程可以线性化。

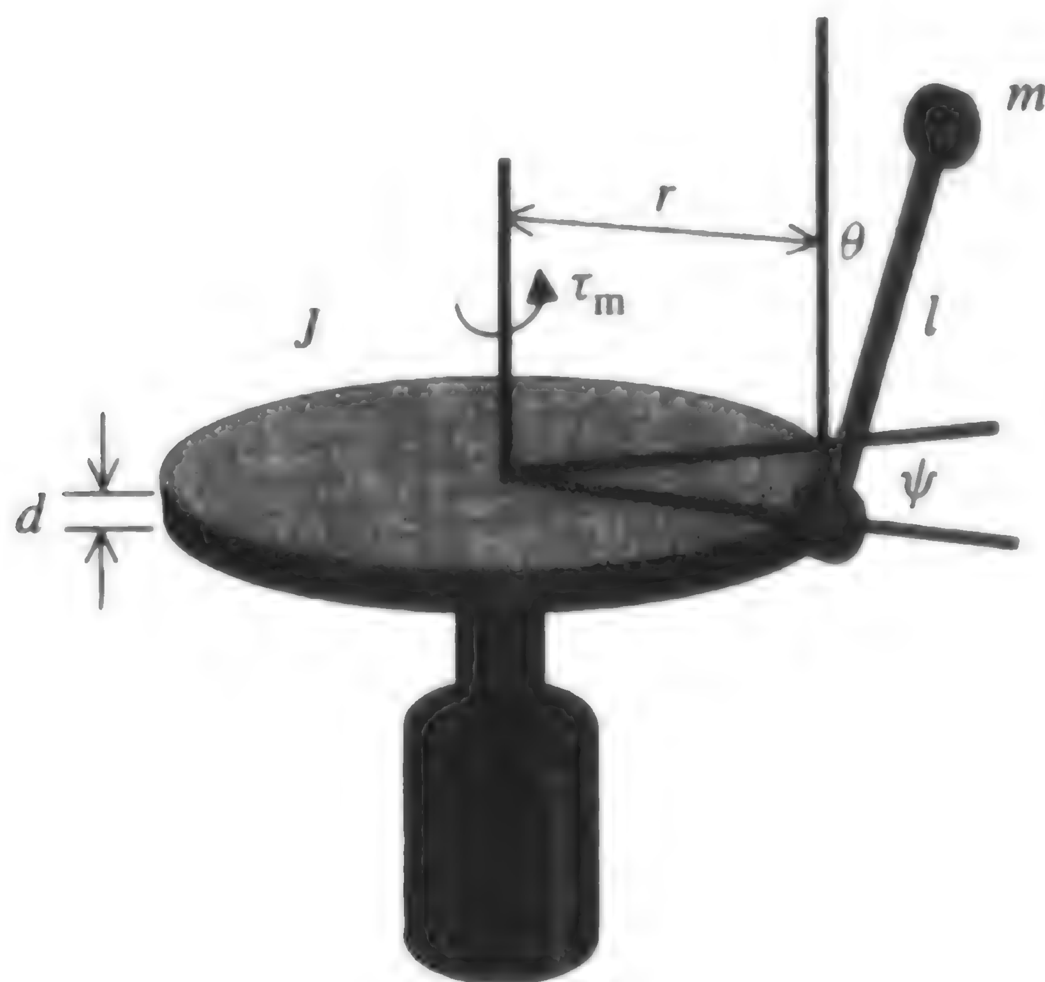


图10.37　圆盘上的倒置摆

定义状态向量 $\boldsymbol{x}$ 为:

$$x_1(t)=\theta(t)\quad x_3(t)=\frac{\mathrm{d}\theta}{\mathrm{d}t}$$

$$x_2(t)=\Psi(t)\quad x_4(t)=\frac{\mathrm{d}\Psi}{\mathrm{d}t}$$

将这些方程代入式(10.14),并假设 θ 和 $\mathrm{d}\theta/\mathrm{d}t$ 非常小,可以忽略掉二阶以上的项,得到:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}=x_3$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t}=x_4$$

$$ml^2\frac{\mathrm{d}x_3}{\mathrm{d}t}+mlr\frac{\mathrm{d}x_4}{\mathrm{d}t}=mglx_1+b_1x_3$$

$$mlr\frac{\mathrm{d}x_3}{\mathrm{d}t}+(J+mr^2)\frac{\mathrm{d}x_4}{\mathrm{d}t}=b_2x_4+\tau_m$$

或矩阵形式:

$$\boldsymbol{M}\dot{\boldsymbol{x}}=\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{W}\boldsymbol{u}$$

其中,

$$\boldsymbol{W}=[0\,0\,0\,1]' \quad \boldsymbol{x}=[x_1 \quad x_2 \quad x_3 \quad x_4]' \quad \dot{\boldsymbol{x}}=\mathrm{d}x/\mathrm{d}t$$

$$\boldsymbol{M}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&ml^2&mlr\\0&0&mlr&J+mr^2\end{bmatrix} \quad \boldsymbol{Q}=\begin{bmatrix}0&0&1&0\\0&0&0&1\\mgl&0&b_1&0\\0&0&0&b_2\end{bmatrix} \quad \boldsymbol{u}=\tau_m$$

对应于 $x(t)=0$ 的垂直位置是系统在 $u=\tau_m=0$ 时的一个平衡点。在垂直位置倒置摆系统的线性化方程可以用函数 **Pendulum** 描述状态空间。系统的输出为摆角 θ 和圆盘角度 Ψ。因此,摆的输出对象是一个单输入双输出系统,可以用下面的方法描述。*Plant*(1,1)是从 τ_m 到 θ 的传递函数,而 *Plant*(2,1)是从 τ_m 到 Ψ 的传递函数。假设摆的长度为 30 cm,摆锤的质量为 0.2 kg,圆盘的半径为 15 cm,圆盘的厚度为 1 cm,且其密度为 2500 kg/m³。系统的摩擦系数设为 0,即 $b_1=b_2=0$。

```
function Plant = Pendulum
l = 0.3;g = 9.81;m = 0.2;r = 0.15;
d = 0.01;rho = 2500;b1 = 0;b2 = 0;
J = 0.25 * pi * rho * d * r^4;
M = [1,0,0,0;0,1,0,0;0,0 m * l^2 m * r * l;0,0,m * r * l,J + m * r^2];
Q = [0,0,1,0;0,0,0,1;m * g * l,0,b1,0;0,0,0,b2];
W = [0;0;0;1];
A = inv(M) * Q;
B = inv(M) * W;
C = [1,0,0,0;0,1,0,0];
D = [0];
Plant = ss(A,B,C,D);
```

通过在 MATLAB 命令窗口中键入:

```
Pole(Pendulum)
```

得到系统极点。显示:

```
0
0
6.8923
-6.8923
```

有两个极点在原点处,另外两个在实轴 ±6.9 rad/s 处。对称于虚轴有一对极点,这表明系统在开环状态下是不稳定的。在 MATLAB 命令窗口中输入:

```
Plant = Pendulum;
tzero(Plant(1,1))
```

结果显示两个零点,也就是从摆的角度 θ 观测到的传输零点。只有第一个输入 τ_m 和第一个输出 θ 的系统可以用矩阵的形式访问,即 *Plant* 的下标(1,1)。这样就有两个零点在极点的正上方。这说明仅根据输出 θ 控制倒置摆会忽略某些动态特性,尤其当摆角未包含足够的信息来表示盘的位置和速度时,这种情况更加明显。因此,当圆盘以恒定的角速度旋转时,角度传感器的输出为零。尽管输出为零,但动态特性仍然存在,有时称之为零动态。这些不能观测的

零动态不能仅通过 θ 的反馈进行稳定。从圆盘角度 Ψ 观测摆的零点可通过在 MATLAB 命令窗口中输入

```
Plant = Pendulum;
tzero(Plant(2,1))
```

获得。由于对象有两个输出,因此可用下标(2,1)获得子系统的第2个输出和第1个输入。从这些结果中可以看出,零点位于 ±5.72,与摆的开环极点 ±6.9 接近。这说明从盘的位置很难看出摆的特性。根据任何一个输出设计而成的单入单出(SISO, single-input single-output)控制器,其运行特性都很差。

取决于两个输出的多入单出(MISO, multiple-input single-output)控制器,其运行特性要比SISO控制器好很多。这将是进一步设计的关键所在。设计图10.38所示的控制器可分为两步:首先对输出 θ 设计控制器,保持摆的垂直,然后围绕这个控制器对 Ψ 设计控制器,保持圆盘在指定位置。

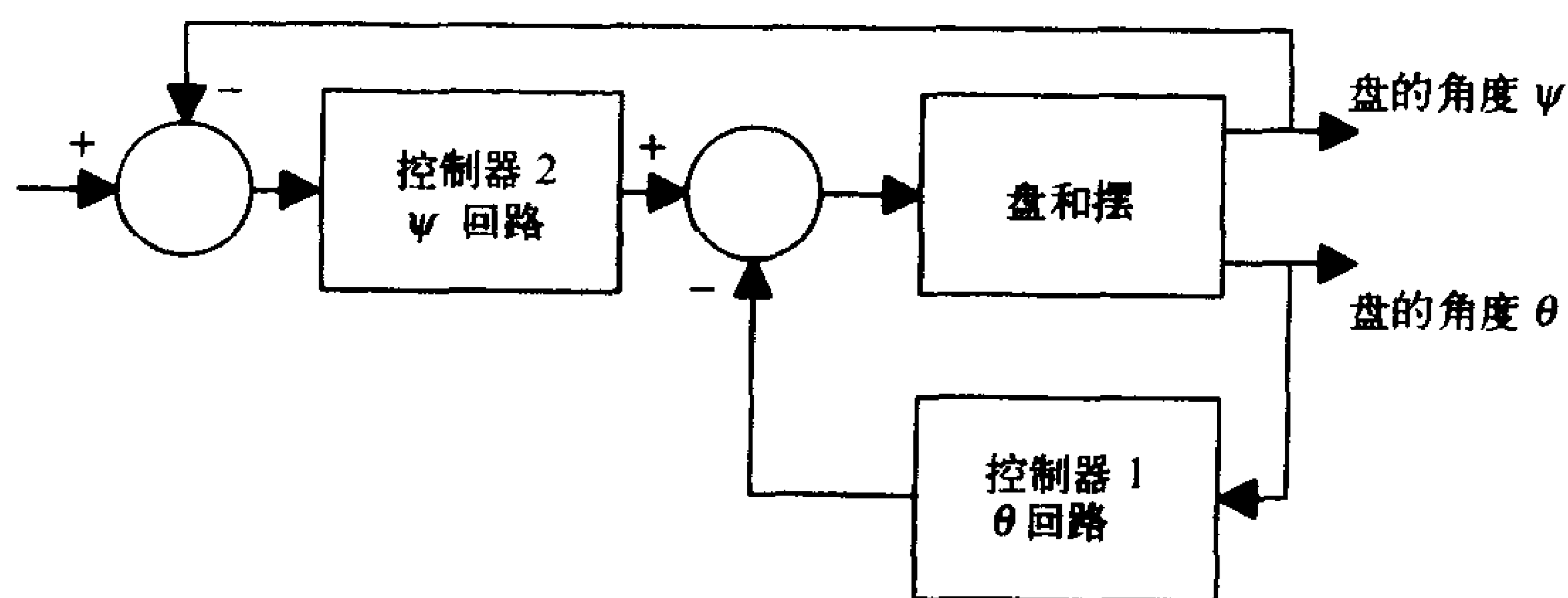


图10.38 倒置摆控制系统的方框图

下面用超前控制器保持摆的垂直,这样的控制器用输出 θ 来确定采用哪种补偿方式。参照位于 ±6.9 处的开环极点,将超前零点放在 −5 处,正好位于稳定极点的右侧,其目的是将不稳定极点移到复平面的左半部分。超前极点位于 −10,因此,

$$C_\theta(s) = -\frac{s+5}{s+10}$$

由于要求为正反馈,故此 $C_\theta(s)$ 中出现负号。

下面的程序用于生成 θ 控制回路的根轨迹图。MATLAB 可自动选择增益范围,但该例中采用在 $0.1 \leqslant \theta \leqslant 10$ 的范围内按对数比例进行等分。

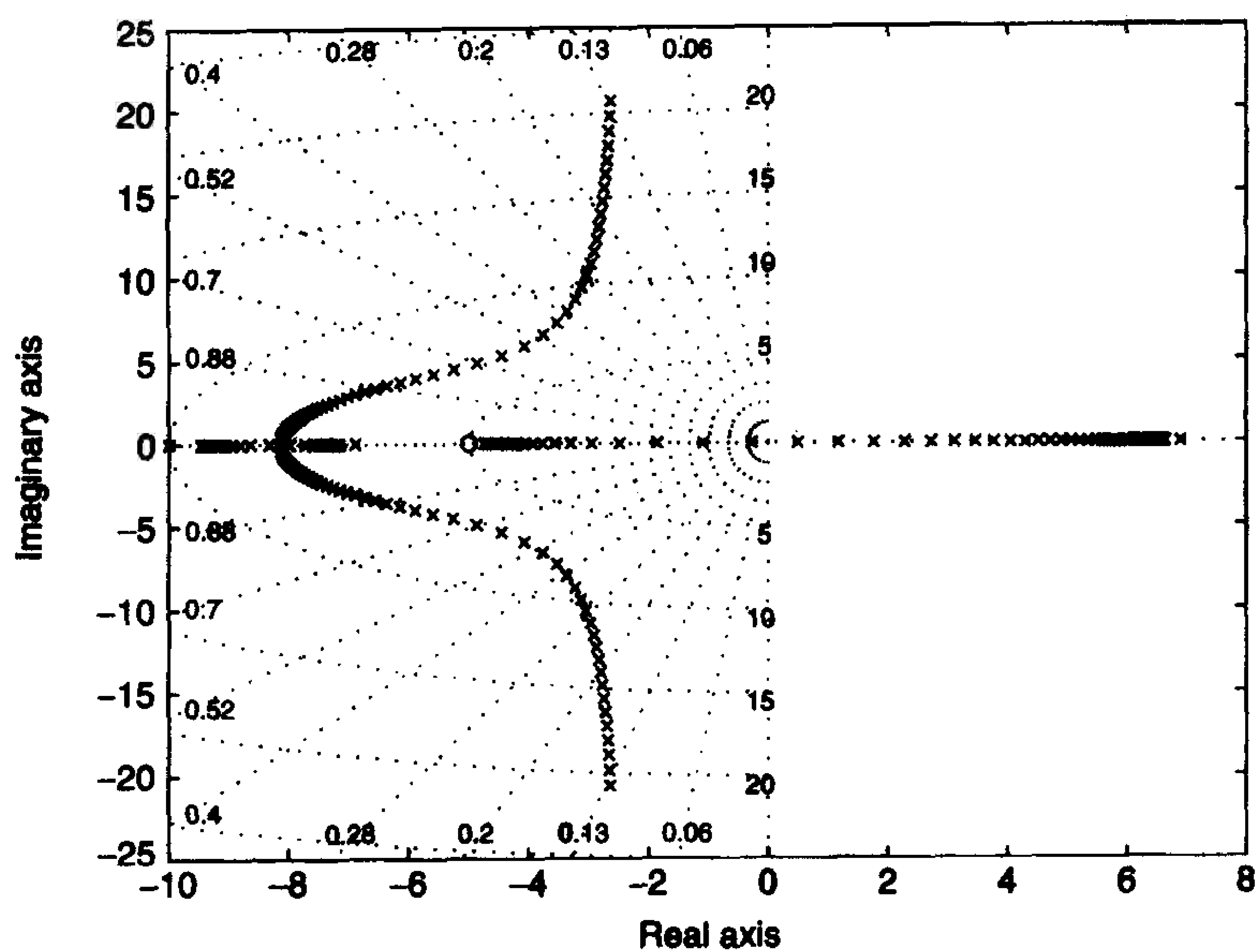
```
Plant = Pendulum;
PlantTheta = minreal(Plant(1,1));
ControlTheta = tf(-1*[1 5],[1,10]);
rlocus(ControlTheta * PlantTheta, logspace(-1,1,60));
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果如图10.39(a)所示。

增益可通过下列程序获得:

```
Plant = Pendulum;
PlantTheta = minreal(Plant(1,1));
```

```
ControlTheta = tf( - 1 * [1,5],[1,10]);
rlocus(ControlTheta * PlantTheta)
rlocfind(ControlTheta * PlantTheta)
```



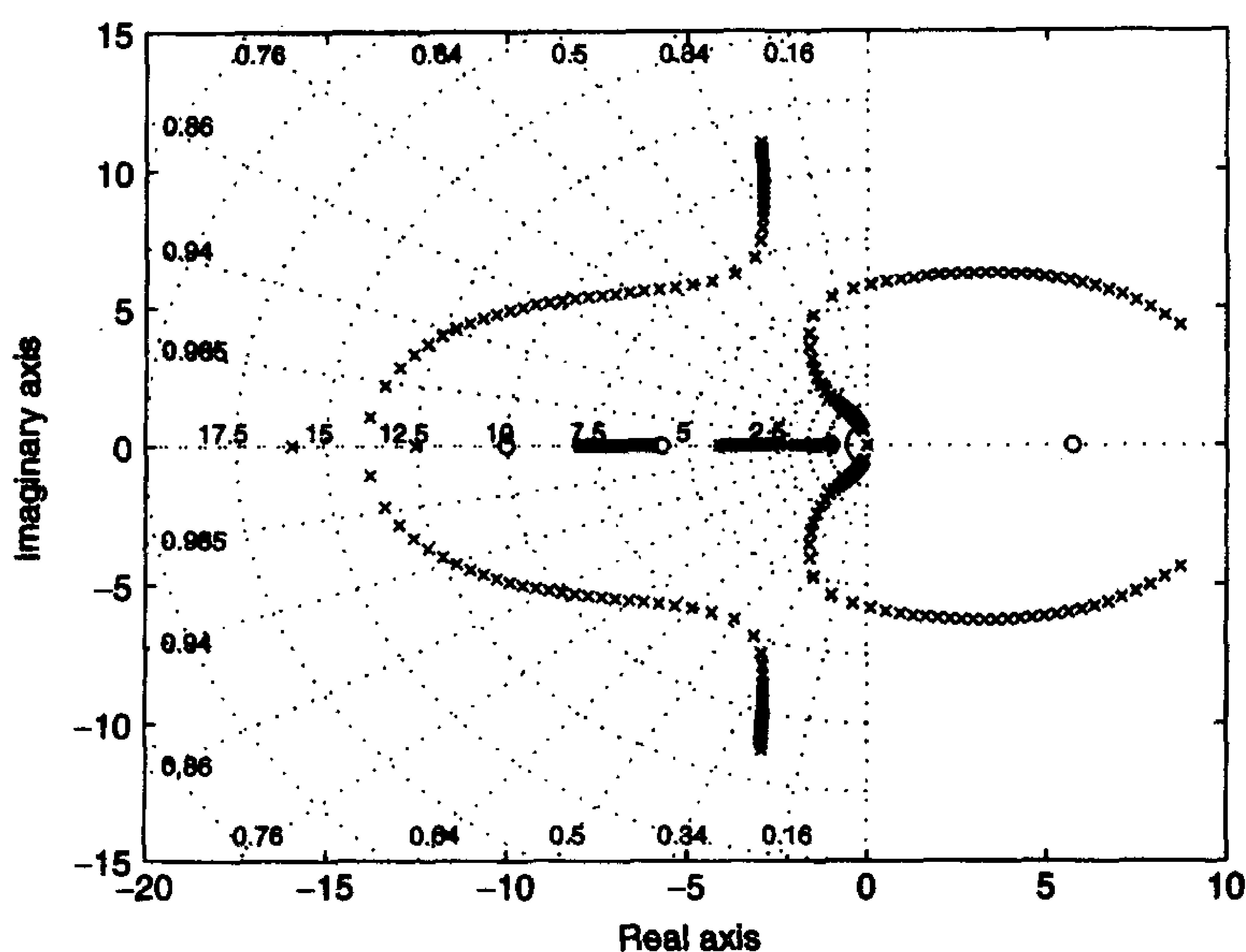


图 10.39　反馈控制回路的根轨迹:(a) θ 控制回路; (b) Ψ 控制回路

将十字线放在虚轴与超前零点之间的实轴上,则可得到使 θ 回路稳定的增益。由下面的程序可得出 θ 控制系统的闭环极点:

```
Plant = Pendulum;
PlantTheta = minreal(Plant(1,1));
```

```
ControlTheta = tf( - 1 * [1,5],[1,10]);
pole(feedback(4 * ControlTheta * PlantTheta,1))
```

程序执行结果显示,闭环系统极点近似为 -4 和 $-3\pm 11i$,系统尽管无阻尼,但却是稳定的。

现在设计圆盘位置 Ψ 的外部反馈回路。外部反馈回路与内部反馈回路相关,因此首先将第一个控制器置入 θ 回路中形成闭环回路系统。可通过下列程序实现该项操作:

```
ControlTheta = tf( - [1,5],[1,10]);
PlanPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
pole(PlantPsi(2,1))
tzero(PlantPsi(2,1))
```

注意,因为对象是 MIMO,而控制器是 SISO,所以必须指定使用哪个输入和哪个输出。结果显示 *PlantPsi* 在原点处还有两个附加的极点,相当于一个双积分器。这就需要将其向左移,目的是产生一个不稳定的零点。倒置摆是非最小相位系统的例子。为了转动圆盘,控制器必须先向相反的方向转动,以保持摆从当前位置运动到期望位置时一直保持垂直。不稳定零点可以吸收双积分器中的一个极点。为了解决这个问题再次使用超前控制器,其零点刚好在复平面的左半平面。下面的程序画出了 Ψ 控制器回路的根轨迹:

```
ControlTheta = tf( - [1,5],[1,10]);
PlantPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
ControlPsi = tf( - [1,1],[1,8]);
k = 0.35 * logspace( - 1,1,60);
rlocus(ControlPsi * PlantPsi(2,1),k);
sgrid;
xlabel('Real axis')
ylabel('Imaginary axis')
```

程序执行结果如图 10.39(b)所示。在 MATLAB 命令窗口键入:

```
Control Theta = tf( - [1,5],[1,10]);
PlantPsi = feedback(Pendulum,4* ControlTheta,[1],[1]);
Control Psi = tf( - [1,1],[1,8]);
rlocus(Control Psi* PlantPsi(2,1))
rlocfind(Control Psi* PlantPsi(2,1))
```

在靠近回路进入复平面的根轨迹线的下半部分放置十字线可得到合适的增益。增益值为 0.3 时,将所有极点放置在左半平面。

最后,控制系统的阶跃响应曲线可由下面的程序算出:

```
ControlTheta = tf( - [1,5],[1,10]);
PlantPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
ControlPsi = tf( - [1,1],[1,8]);
[y,t] = step(feedback(0.3 * ControlPsi * PlantPsi,1,[1],[2]));
plot(t,y(:,1),'k-',t,y(:,2),'k--');
xlabel('Time');
ylabel('Step response');
legend('\theta(t)','\psi(t)');
```

阶跃响应曲线如图 10.40 所示。结果揭示了控制器和被控对象具有非最小相位系统的特性。

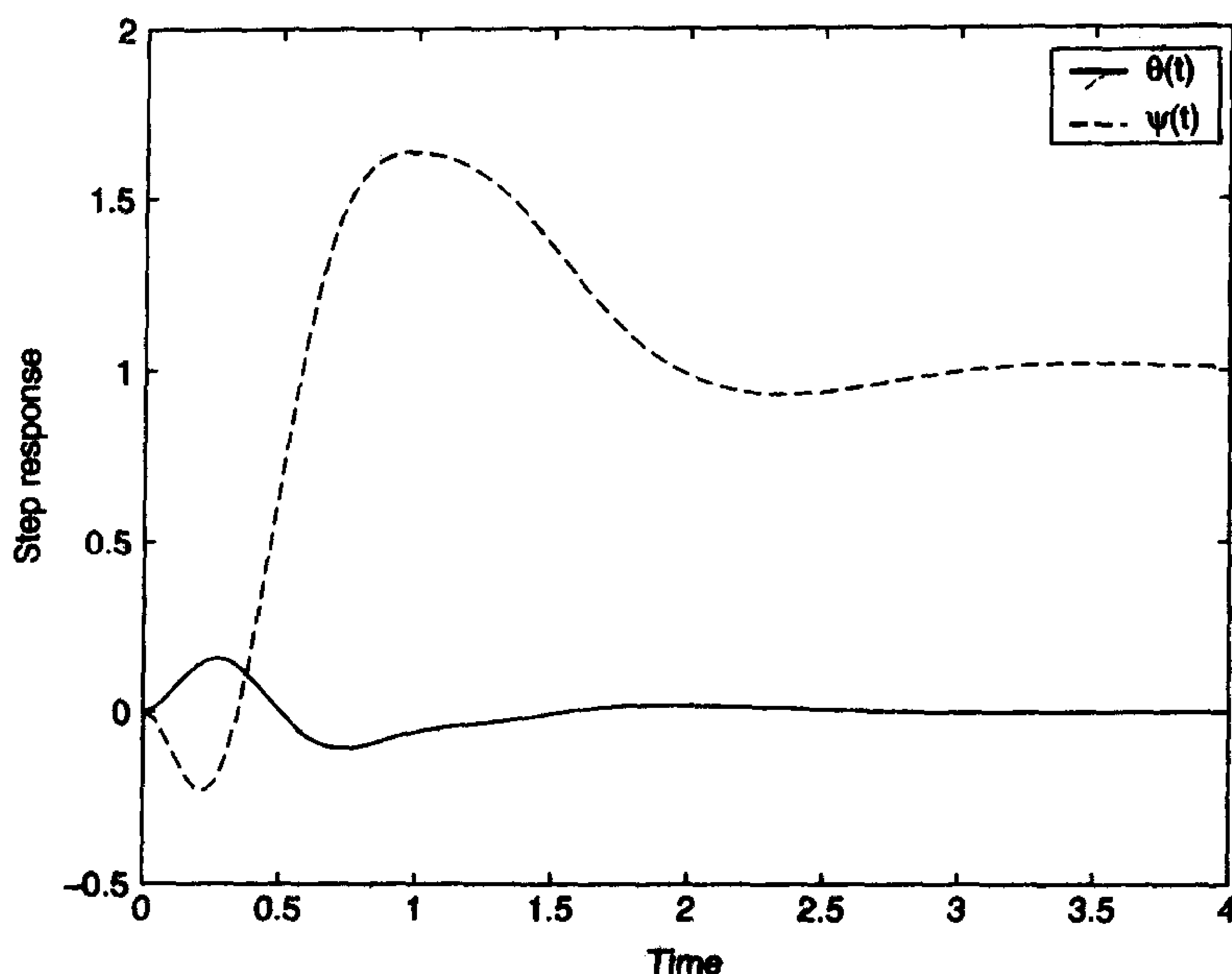


图 10.40 倒置摆的阶跃响应曲线

10.5.4 磁悬浮飞轮的控制

研究图 10.41 所示的磁悬浮飞轮系统。电磁线圈用来使飞轮悬浮,所以飞轮能够以很高的速度运行,而且没有摩擦损耗。控制系统的目标是保持飞轮的悬浮状态。与 10.5.2 节描述的磁悬浮系统一样,系统实质上是不稳定的。输出为 4 个可以测量的距离。在可测量的惯性参照系中,这些距离对应于轴的顶部和底部 x 和 y 的位置。4 个线圈电流控制轴的磁场,每个线圈都配有传感器。运动的线性化方程为:

$$\frac{d^2 x_{cm}}{dt^2} = \frac{f_1 + f_3}{m}$$

$$\frac{d^2 y_{cm}}{dt^2} = \frac{f_2 + f_4}{m}$$

$$\frac{d^2 \phi}{dt^2} = -\beta\omega \frac{d\Psi}{dt} + \frac{h}{J_{xx}}(f_4 - f_2)$$

$$\frac{d^2 \Psi}{dt^2} = \beta\omega \frac{d\phi}{dt} + \frac{h}{J_{xx}}(f_1 - f_3)$$

其中,x_{cm}和 y_{cm}是在可测量的惯性参照系中飞轮的重心位置,在惯性参照系中,(ϕ, Ψ)用旋转(ϕ)和倾角(Ψ)给出飞轮参照系的方位和偏转方向①,m 为飞轮的质量,J_{xx}是相对于非自旋轴的飞轮的转动惯量,$\beta = J_{zz}/J_{xx}$,h 为重心到调节器的距离。输入 f_i 为磁性轴承的作用力并遵循以下关系式:

$$f_i = k_1 y_i + k_2 u_i$$

① 参见 R. M. Murray, X. Li 和 S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, FL, 1994 年。

其中，y_i 是从飞轮到调节器的距离，也是系统的输出。轴是由缠有线圈的永磁铁构成的。负恢复常数 k_1 由永磁铁确定，增益 k_2 由线圈中的电流 u_i 产生的磁场确定。轮的运行速度为 ω rad/s。

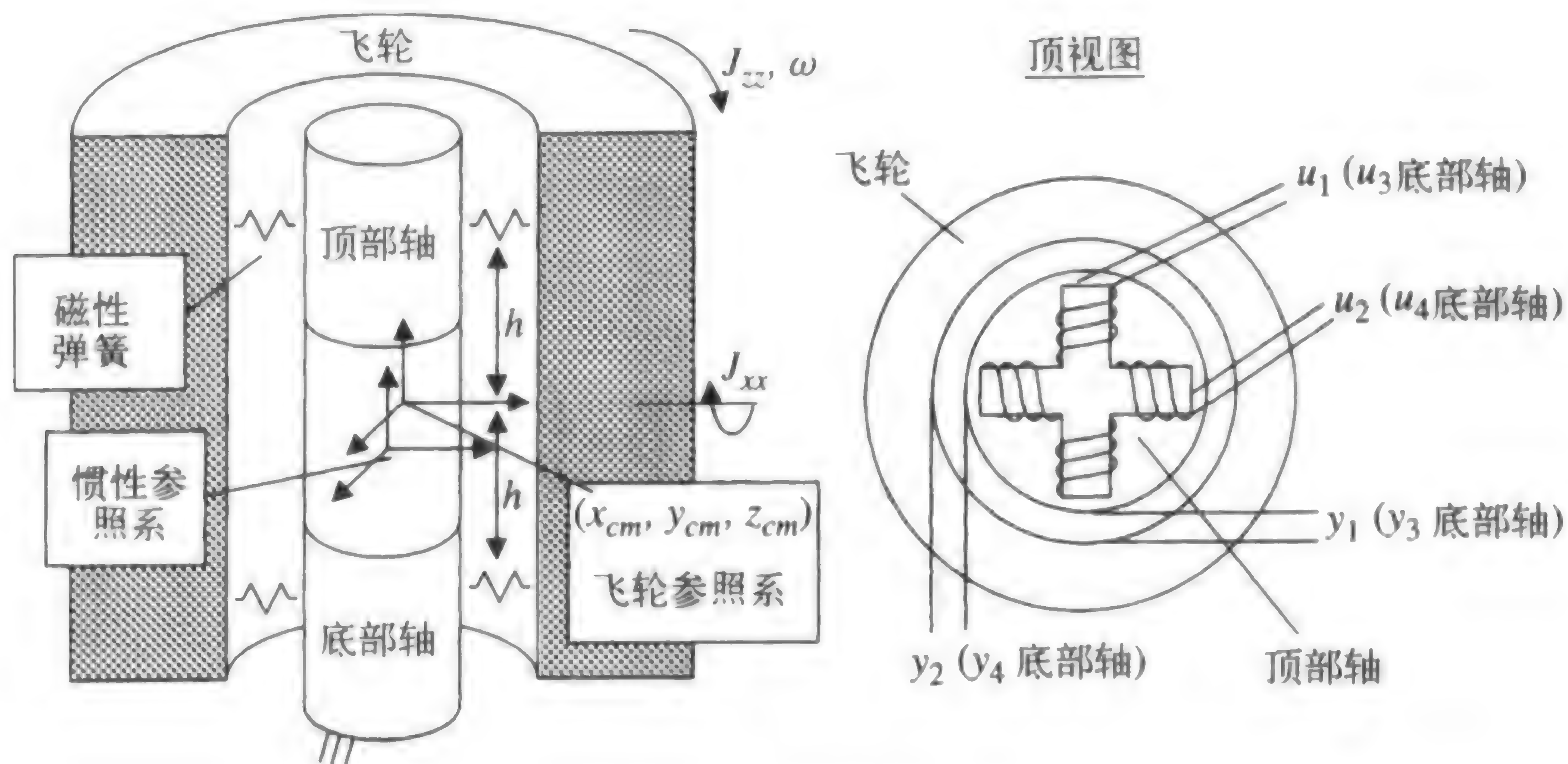


图 10.41　磁悬浮飞轮

对于很小的 ϕ 和 Ψ 值，y_i 由下式给出：

$$
\begin{aligned}
y_1 &= x_{cm} + h\Psi \\
y_2 &= y_{cm} - h\phi \\
y_3 &= x_{cm} + h\Psi \\
y_4 &= y_{cm} - h\phi
\end{aligned}
$$

如果令：

$$q(t) = [x_{cm}, y_{cm}, \phi(t), \Psi(t)]'$$

且

$$u(t) = [u_1\ u_2\ u_3\ u_4]'$$

则线性化方程可以写为：

$$
\begin{aligned}
\ddot{q} &= \omega P_a \dot{q} + k_1 B_a C_a q + k_2 B_a u \\
y &= C_a q
\end{aligned}
$$

其中，

$$
\boldsymbol{P}_a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta \\ 0 & 0 & -\beta & 0 \end{bmatrix}
$$

$$
\boldsymbol{B}_a = \begin{bmatrix} 1/m & 0 & 1/m & 0 \\ 0 & 1/m & 0 & 1/m \\ 0 & -h/J_{xx} & 0 & h/J_{xx} \\ h/J_{xx} & 0 & -h/J_{xx} & 0 \end{bmatrix} \tag{10.15}
$$

$$C_a=\begin{bmatrix}1&0&0&h\\0&1&-h&0\\1&0&0&-h\\0&1&h&0\end{bmatrix}$$

若设:

$$z=\begin{bmatrix}q\\\dot{q}\end{bmatrix}$$

可得到下面的矩阵方程:

$$\dot{z}=\begin{bmatrix}0&I\\k_1B_aC_a&\omega P_a\end{bmatrix}z+\begin{bmatrix}0\\k_2B_a\end{bmatrix}u$$

$$y=[C_a\quad 0]z+[0]u$$

其中,I 为单位矩阵。

下面的函数 **Flywheel** 可生成线性化的模型,作为旋转速度的函数,转速单位为 r/m。此模型将应用于下面的设计计算中。如果该函数没有输入参数,则假定其运行速度为 0 r/m。对于本例中的飞轮,x 和 y 轴的转动惯量为 1.563×10^{-4} Nm·s^2,z 轴的转动惯量为 1.141×10^{-4} Nm.s^2,无量纲形式的因子 β 近似为 1。飞轮的质量为 340 gm,重心的高度为 3 cm。线圈系数 $k_1=4.8\times10^4$ N/m,$k_2=3.75$ N/A。

```
function Plant = Flywheel(rpm)
if nargin < 1, rpm = 0; end;
Jxx = 1.563e-4; Jzz = 1.14le-4;
beta = Jzz/Jxx; m = 0.34;
h = 0.03; k1 = 4.8e4; k2 = 3.75;
omega = rpm/60 * 2 * pi;
Pa = [zeros(1,4); zeros(1,4); 0,0,0, -beta; 0,0,beta,0];
Ba = [1/m,0,1/m,0; 0,1/m,0,1/m; 0, -h/Jxx,0,h/Jxx; h/Jxx,0, -h/Jxx,0];
Ca = [1,0,0,h; 0,1, -h,0; 1,0,0, -h; 0,1,h,0];
A = [zeros(4), eye(4); k1 * Ba * Ca, omega * Pa];
B = [zeros(4); k2 * Ba];
C = [Ca, zeros(4)];
D = [zeros(4)];
Plant = ss(A,B,C,D);
```

开环极点随 r/m 变化。运行下面的程序可画出一组在典型运行速度条件下的开环极点:

```
rpm = [0:100:16000, 16100:10:20000];
result = zeros(8, length(rpm))
for j = 1:length(rpm)
  result(:, j) = pole(Flywheel(rpm(j)));
end
plot(real(result(:1)), imag(result(:,1)), 'x');
hold on;
for j = 1:8;
  x = real(result(j,:));
  y = imag(result(j,:));
  plot(x,y);
end;
grid;
```

```
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果如图 10.42 所示。有两组对称于虚轴的极点(没有零点),一组位于 ±530 rad/s(85 Hz),与运行速度无关,是由直线运动模型而引起的;另一组位于 ±740 rad/s(120 Hz)受陀螺耦合影响,是由两个旋转模型而引起的。

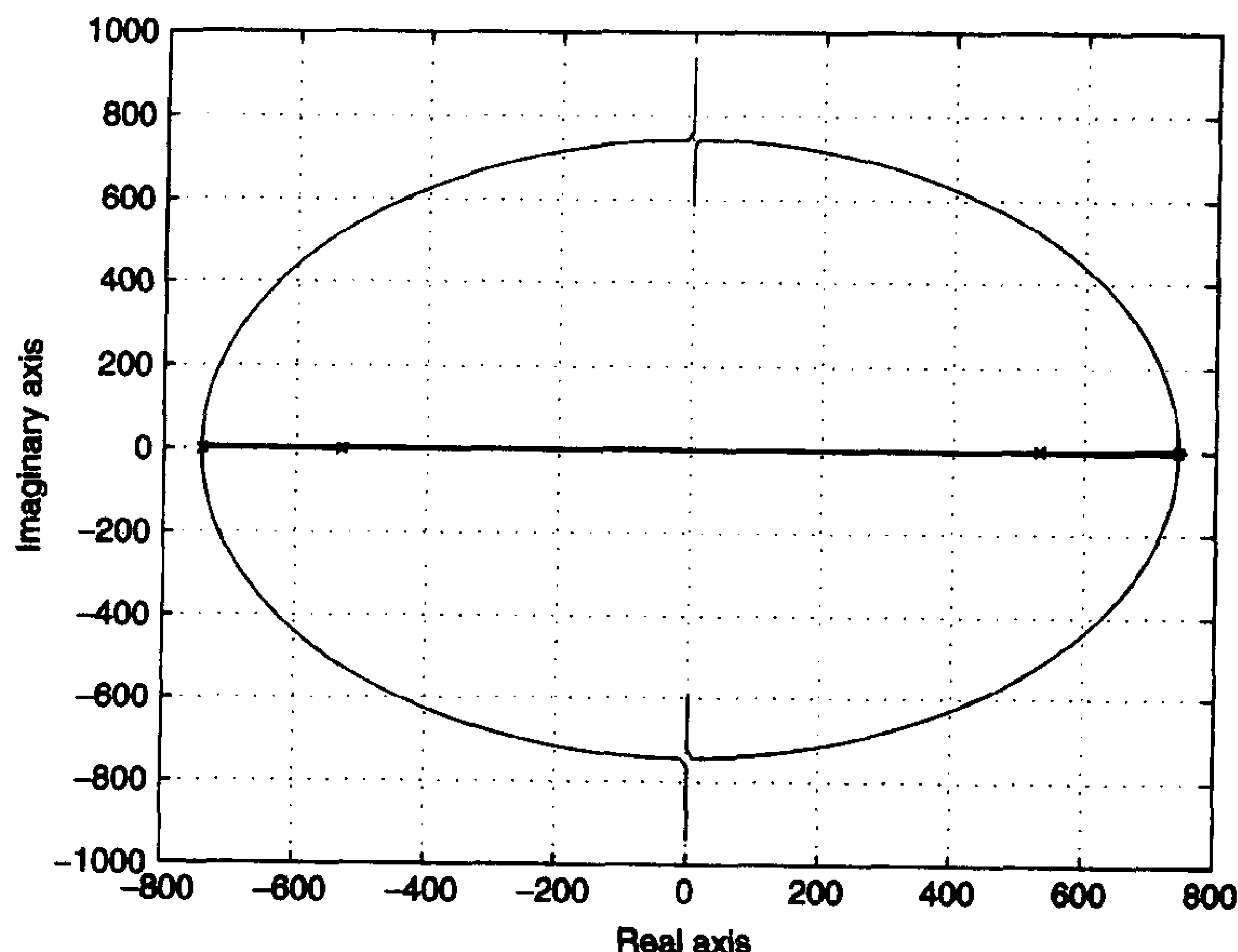


图 10.42　从 0 到 20 000 r/m 的运行速度函数的根轨迹

总之,有两个不稳定极点和两个稳定极点,有一半的极点随运行速度改变。查看式(10.15)可以看出,在 $\omega = 0$ r/m 处,P 不影响结果。这说明飞轮在转速为 0 rad/m 时可实现静态解耦,并形成两个旋转、两个平移的 SISO 系统。因此,像根轨迹法这样的传统超前控制方法可用来设计稳定的控制器。

下面来研究矩阵 $\boldsymbol{T}$ 在多个输入通道之间形成信号的和与差。

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

注意,矩阵 $\boldsymbol{T}$ 使得输入/输出矩阵方程(10.15)对角化,如下所示:

$$\boldsymbol{BT} = \begin{bmatrix} 2/m & 0 & 0 & 0 \\ 0 & 2/m & 0 & 0 \\ 0 & 0 & 2h/J_{xx} & 0 \\ 0 & 0 & 0 & 2h/J_{xx} \end{bmatrix}$$

$$\boldsymbol{T}^{-1}\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & h & 0 \\ 0 & 0 & 0 & h \end{bmatrix}$$

用矩阵 T 完成静态解耦之后，即可进行 4 个 SISO 对象的超前设计。图 10.43 说明在实际中如何插入矩阵 T。下面的程序可产生解耦后的 SISO 对象，并计算出结果的根轨迹：

```
T=[1,0,0,1;0,1,-1,0;1,0,0,-1;0,1,1,0];
decoupFly=inv(T)*Flywheel(0)*T;
transFly=minreal(decoupFly(1,1));
rotFly=minreal(decoupFly(3,3));
Lead=tf([1,400],[1,1000]);
rlocus(Lead*transFly);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
figure(2);
rlocus(Lead*rotFly);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果如图 10.44 所示。

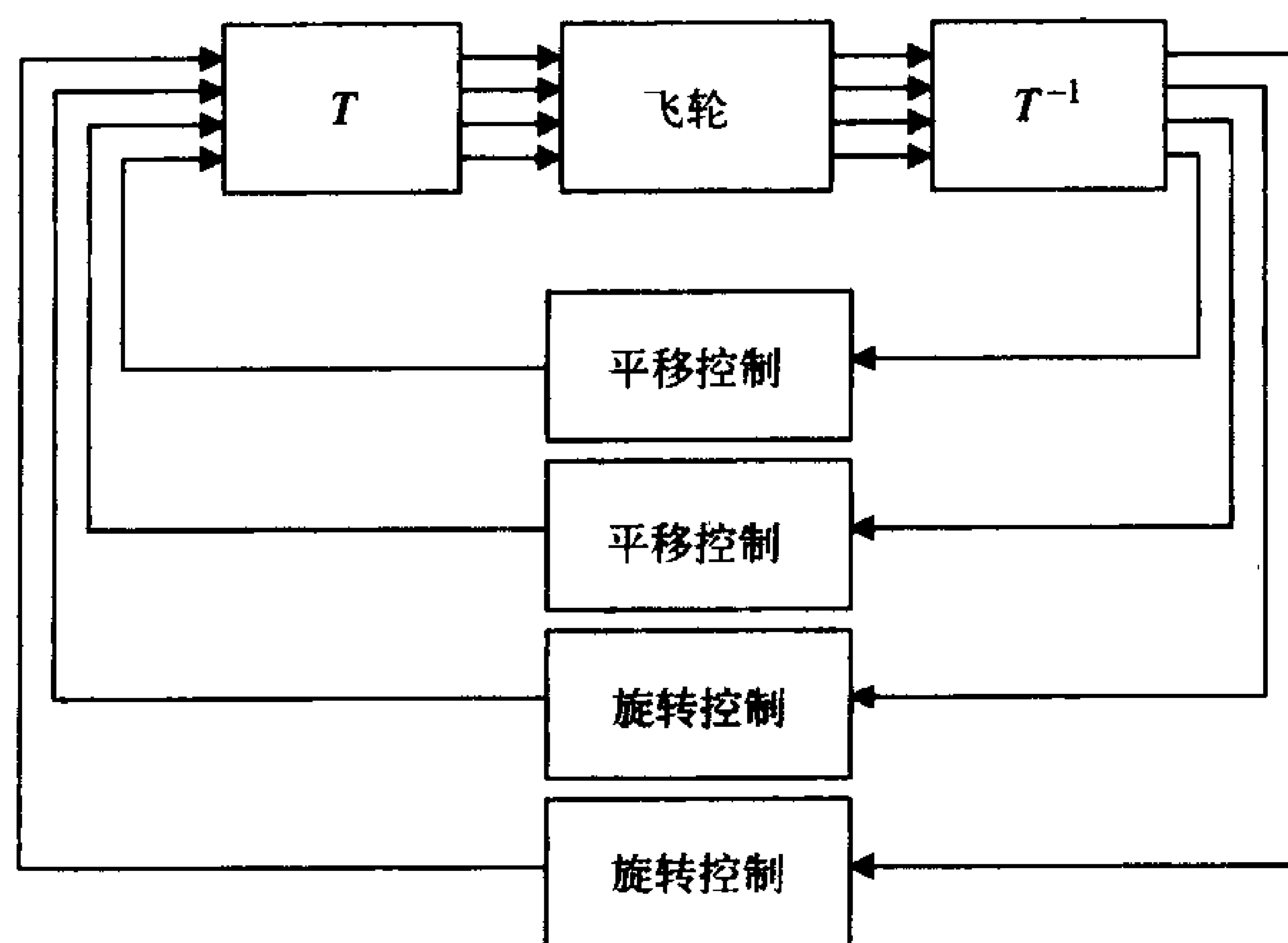


图 10.43 飞轮的静态解耦

程序

```
T=[1,0,0,1;0,1,-1,0;1,0,0,-1;0,1,1,0];
decoupFly=inv(T)*Flywheel(0)*T;
transFly=minreal(decoupFly(1,1));
Lead=tf([1,400],[1,1000]);
rlocus(Lead*transFly)
rlocfind(Lead*transFly)
```

用于寻找直线运动系统的稳定增益，并且程序

```
T=[1,0,0,1;0,1,-1,0;1,0,0,-1;0,1,1,0];
decoupFly=inv(T)*Flywheel(0)*T;
rotFly=minreal(decoupFly(3,3));
Lead=tf([1,400],[1,1000]);
rlocus(Lead*rotFly)
rlocfind(Lead*rotFly)
```

用来寻找旋转系统的稳定增益。通过设计,用相同增益值(5)可同时满足系统中的平移和旋转两个组件的要求。既然平移和旋转控制的增益是相同的,则控制器也是相同的。由于矩阵 T 已折算到了控制器的传递函数之中,所以实现时不需要再进行解耦。这使得实现过程更加容易。

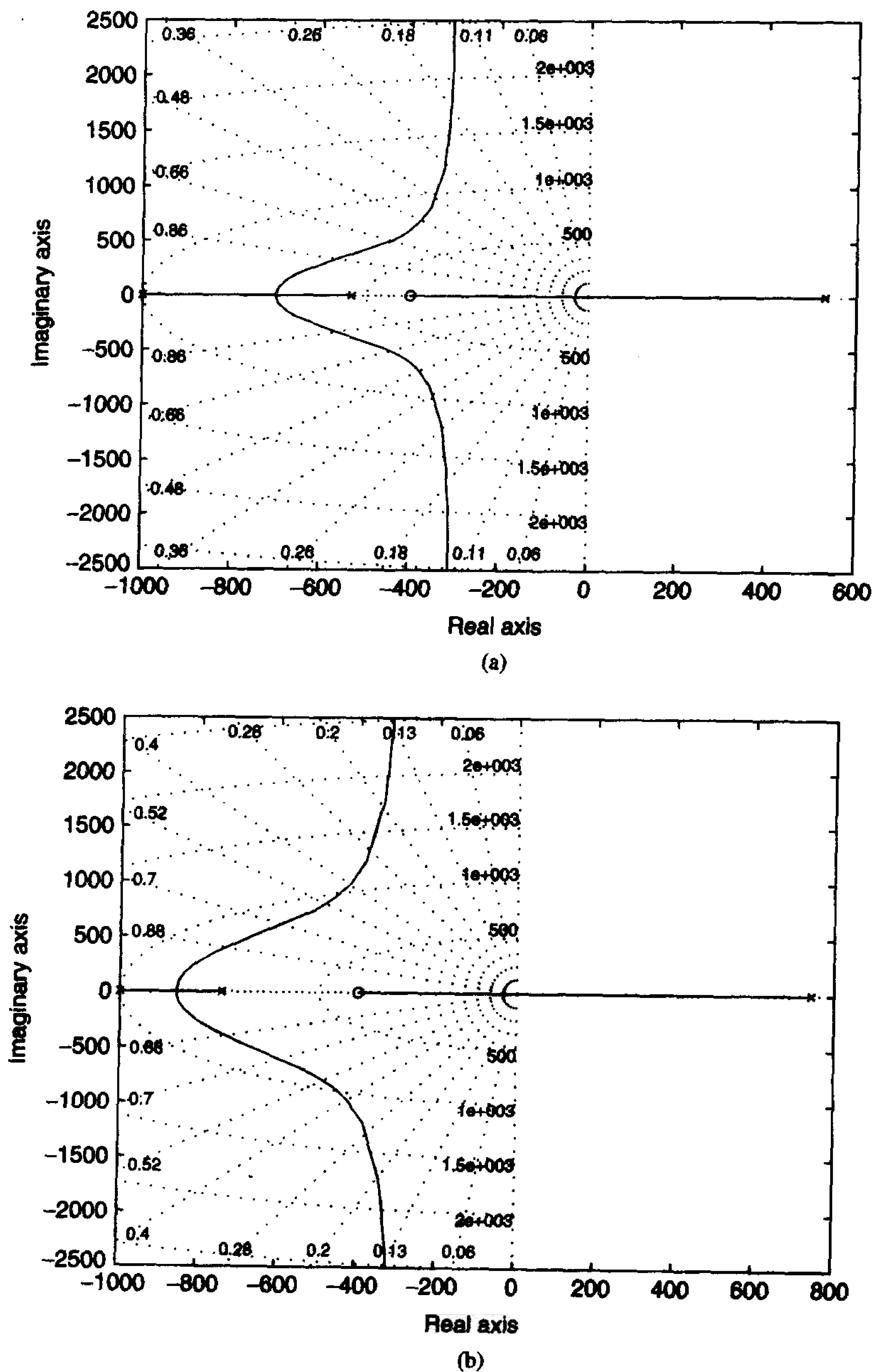


图 10.44 控制回路的根轨迹:(a) 平移运动;(b) 旋转运动

现在为飞轮设计 LQG 控制器。设计包含选择适当的代价函数 R。下面的程序在 $10^{-10} \leqslant R \leqslant 10^{-6}$ 时计算最优闭环极点。程序执行结果如图 10.45 所示。

```
[A,B,C,D] = ssdata(Flywheel);
clPoles = [];
R = logspace( - 10, - 6,60);
for i = 1:length(R)
 [K,S,E] = lqr(A,B,C' * C,R(i) * eye(4));
 clPoles = [clPoles,E];
end;
plot(real(clPoles),imag(clPoles),'kx');
sgrid;
ylabel('Imaginary axis');
xlabel('Real axis');
text( - 700,0,'R = 10^{ - 10}');
text( - 1000,1000,'R = 10^{ - 6}');
```

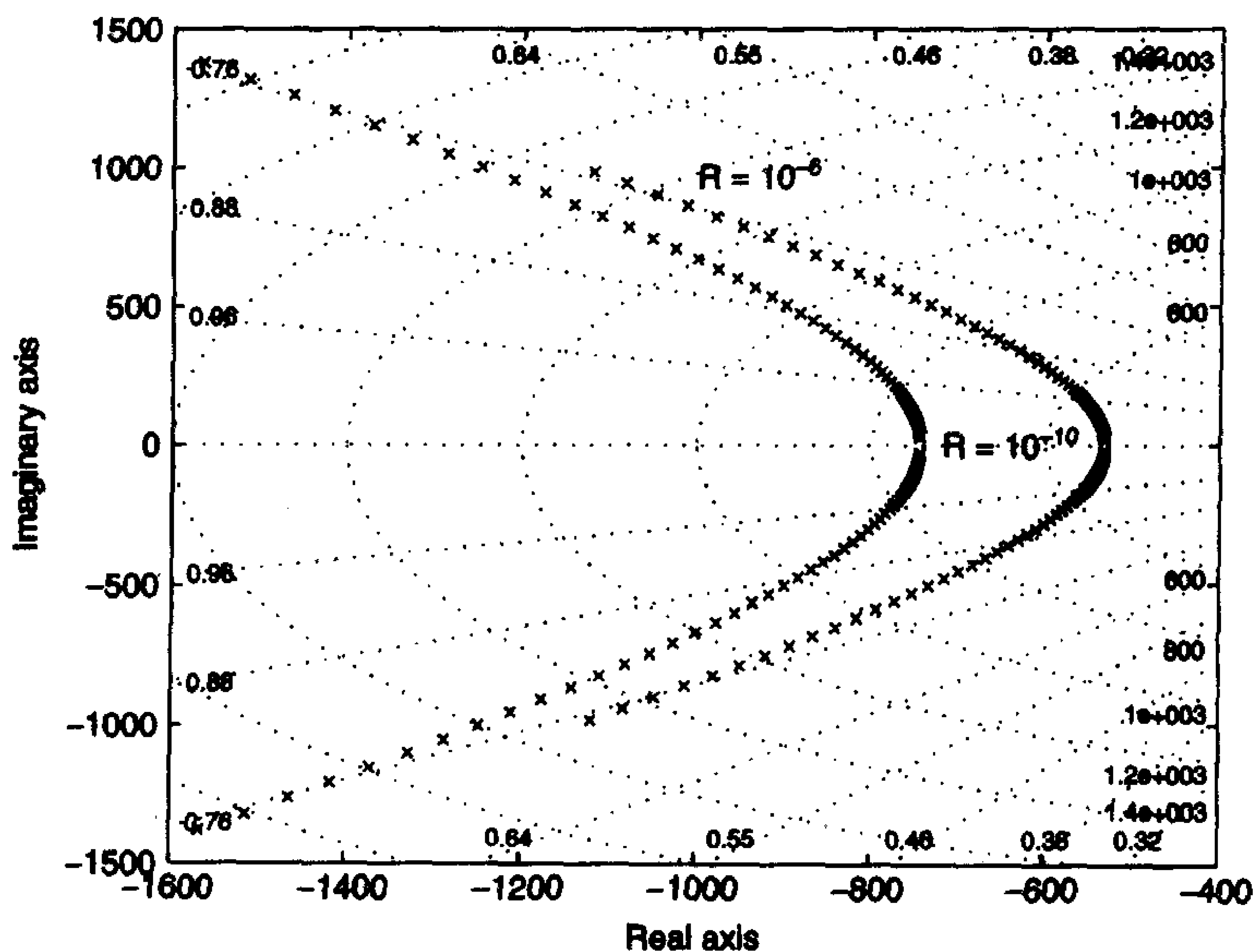


图 10.45　$10^{-10} \leqslant R \leqslant 10^{-6}$时飞轮的最优根轨迹

为比较两种不同的控制方法,在相同的初始条件下,下面的程序计算了两个不同控制器的响应。飞轮略微上倾 $\phi(0) = 0.001$ r,其他初始条件均为零。尽管控制器按 0 r/m 设计,但飞轮的转速仍然为 10 000 r/m。基于系数 $R = 10^{-6}$设计 LQG 控制器。

```
Control = 4e4 * eye(4) * tf([1,400],[1,1000]);
x0 = zeros(12,1);x0(3) = le - 3;t = linspace(0,0.25,1000);
yl = initial(feedback(Flywheel(10000),Control),x0,t);
[A,B,C,D] = ssdata(Flywheel);
K = lqr(A,B,C' * C,le - 6 * eye(4));
L = (lqr(A',C',B * B',le - 6 * eye(4)))';
ControlSS = reg(Flywheel,K,L);
x0 = zeros(16,1);x0(3) = le - 3;
ys = initial(feedback(Flywheel(10000),ControlSS, + 1),x0,t);
plot(1000 * yl(:,1),1000 * yl(:,2),'k-',1000 * ys(:,1),1000 * ys(:,2),'k--');
grid;
xlabel('x upper bearing');
ylabel('y upper bearing');
legend('Lead Control','LQG Control','Location','South West')
```

程序执行结果如相位图 10.46 所示。两个控制器都具有相同的控制性能使飞轮保持悬浮。

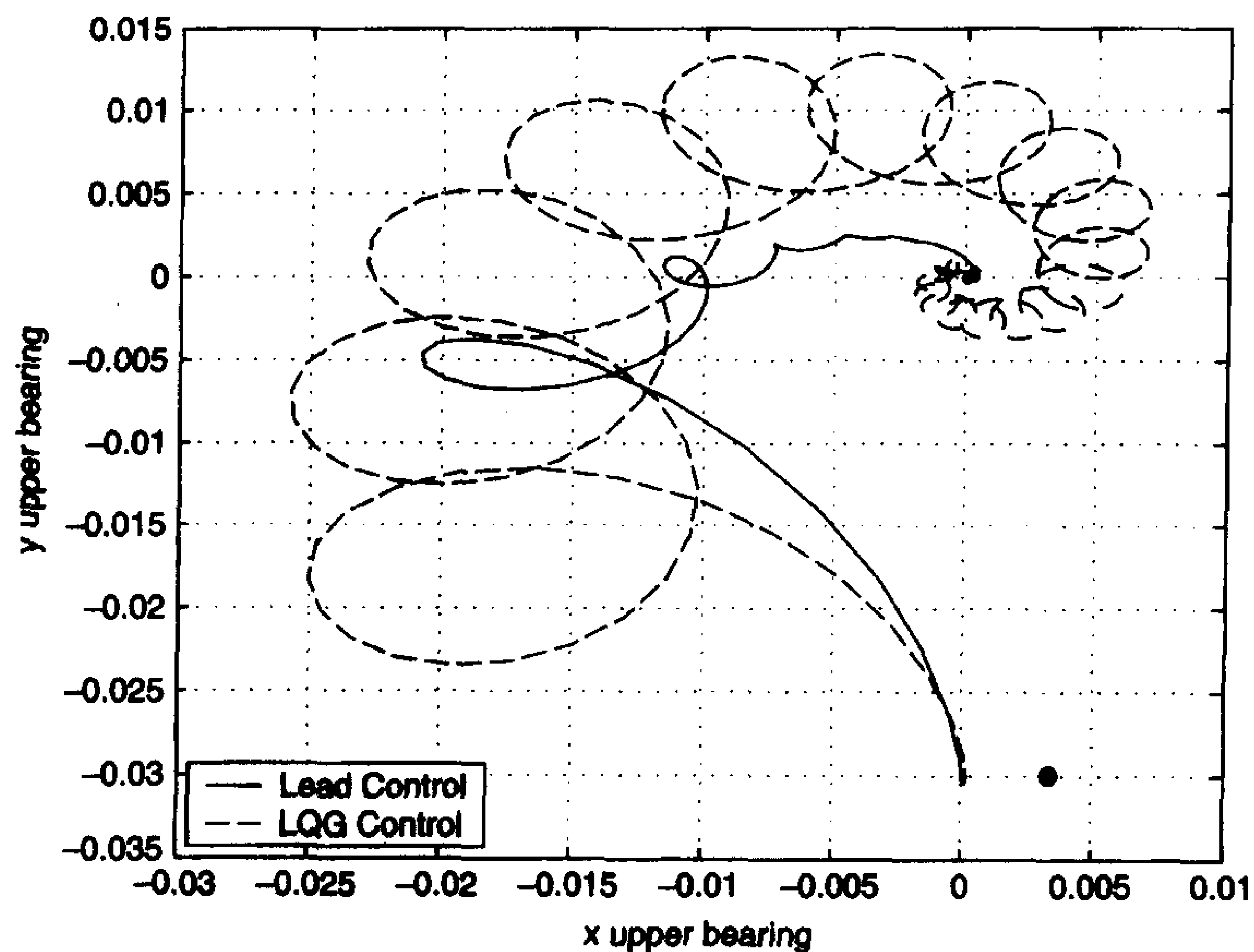


图 10.46　状态空间和超前控制飞轮的相位响应

10.6　应用函数一览

本章介绍的应用函数一览如表 10.1 所示。

表 10.1　第 10 章应用的 MATLAB 函数

MATLAB 函数	说明
acker	单输入系统的极点配置
alpha	设置对象的透明属性
arx *	估算 ARX 模型参数
bode	频率响应波特图
c2d	离散化连续时间系统
compare *	比较测量输出值与模型输出值
connect	由方框图获得状态空间模型
conv	多项式卷积
dcgain	直流增益
detrend *	从输出/输入数据中删除趋向值
feedback	用负反馈连接两个线性非时变模型
iddata *	将输入/输出数据封装到 iddata 对象
ident *	打开图形用户接口(GUI)
ift	两个线性非时变(LTI)模型的 Redheffer star 乘积
impulse	线性非时变(LTI)模型的脉冲响应
initial	状态空间模型的零输入响应
lqe	连续系统的卡尔曼(Kalman)估计器设计
lqr	连续系统的线性二次调节器设计

(续表)

MATLAB 函数	说明
lsim	线性非时变(LTI)模型对任意输入的响应
margin	增益余度、相位余度和对应的转折频率
mineral*	取消零极点的最小实现
nyquist	线性非时变(LTI)模型的奈奎斯特(Nyquist)图
parallel	两个线性非时变(LTI)模型的并联
place	极点配置
pole	线性非时变(LTI)系统的极点
reg	给定状态反馈和估计器增益的调节器模板
rlocfind	查询给定根集的根轨迹增益
rlocus	根轨迹
series	两个线性非时变(LTI)模型的串联
sgrid	具有阻尼因数常数和固有频率的 s 平面栅格
simulink	启动 SIMULINK
ss	状态空间模型
ssbal	使用对角相似变换平衡状态空间
ssdata	访问状态空间模型数据
step	线性非时变(LTI)系统的阶跃响应
tf	传递函数模型
tfdata	访问传递函数数据
tzero	线性非时变(LTI)系统的传输零点
zgrid	具有阻尼因数常数和固有频率的 z 平面栅格
zpk	零点－极点－增益模型

* 系统辨识工具箱

练习

10.1 悬浮系统[①] 如图10.47所示，输入 $y(t)$ 为路面的水平位置，输出 $x(t)$ 为 m_1 的绝对位置。系统的传递函数为：

$$\frac{y(s)}{r(s)}=\frac{sbk_2+k_1k_2}{m_1m_2s^4+b(m_1+m_2)s^3+k_1(m_1+m_2)s^2+k_2m_1s^2+k_2bs+k_1k_2}$$

假设 $m_1=500$ kg, $m_2=100$ kg, $b=1000$ Ns/m, $k_1=2000$ N/m, $k_2=10^4$ N/m。

a. 板式路面可近似为：

$$r(t)=\epsilon\ \sin\omega t$$

当 $y(t)$ 的振幅是 $r(t)$ 振幅的 10% 时，确定 ω 值。

b. 在实际中，汽车的质量增加一倍。在 500 kg $\leqslant m_1 \leqslant$ 1000 kg 的范围内，将波特图中的幅值作为 m_1 的函数，生成网格表面图。

c. ω 作为 b 和 m_1 的函数，当道路变化 $r(t)$ 减为 90% 时，画出 ω 的值。

---

① 参见 U. Ozguner, H. Goktas 和 H. Chan, "Automoative Suspension Control Through a Computer Communication Network," *Proceedings of 1st IEEE Conference on Control Application*, 第 2 卷, 1992 年, 第 895 ~ 900 页。

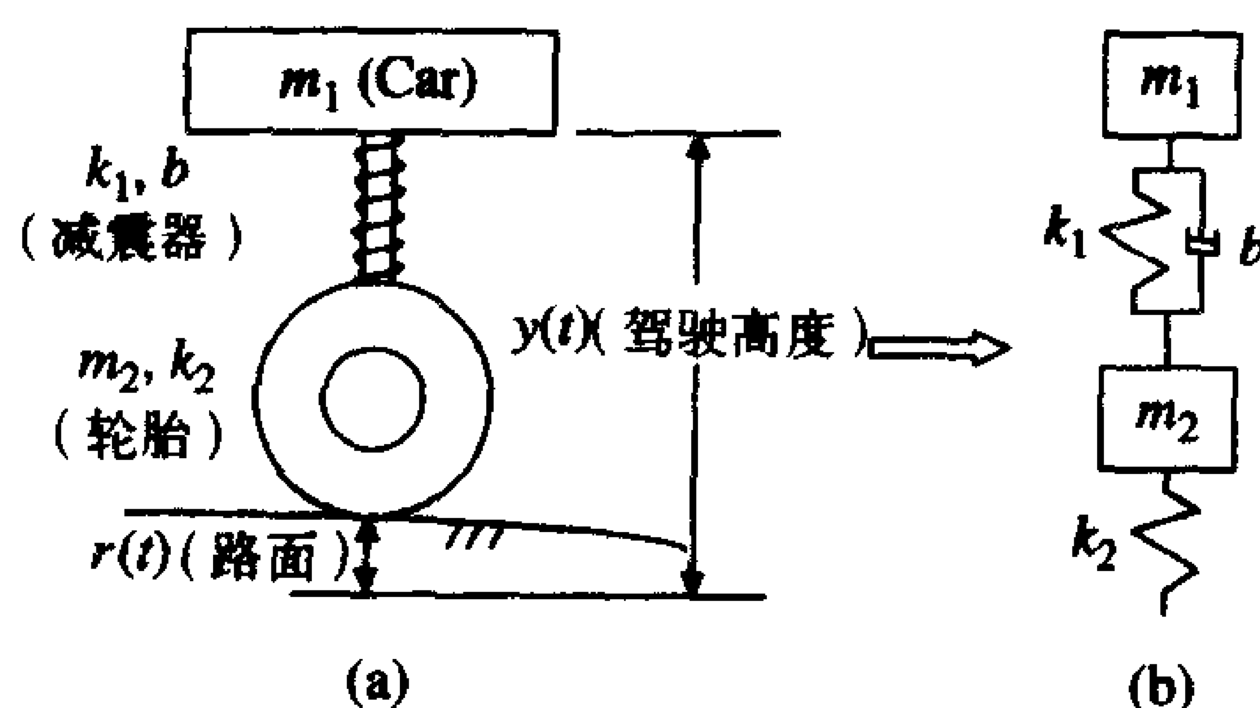


图 10.47 (a) 简化的车轮悬浮模型;(b) 质量和弹簧等效系统

10.2 构建一个 5 阶系统,其极点等间隔分布在左半平面直径为 $2\pi k$ 的圆上,5 个零点在右半平面的对称位置。令直流增益为 1,且 $k=1$。

a. 将 5 阶系统与一个简单的一阶对象串联,一阶对象极点为 -1,直流增益为 1。在同一张图上,画出时间为 4 s 的系统阶跃响应曲线。

b. 计算超调量和上升时间。

c. 当 $k=0.5$ 及 $k=2.0$ 时,重复(a)和(b)两步。观察随着 k 的增加,响应延迟增加的情况。

d. 当有 3 个和 7 个等分极点时,重复(a)和(b)两步。观察随着极点数目的增加,振荡增大的情况。

10.3 用下面的简单常态模型设计控制器[①]:

$$G_0(s)=\frac{1}{s+1}$$

对象不能精确地模型化。对应下面给出的几个模型,画出常态模型 $G_0(s)$的开环阶跃响应和相应对象模型的开环阶跃响应。画出采用比例误差控制,且增益为 20 的闭环阶跃响应。最大时间为 2 s。注意,尽管开环响应差别很大,但闭环响应几乎相同。

a. $$G_1(s)=\frac{3.7}{0.75s+0.6}$$

b. $$G_2(s)=\frac{1.63}{0.94s+0.92}$$

c. $$G_3(s)=\frac{0.7s^2+7s+17}{s^3+2s^2+5.2s+4}$$

10.4 研究 10.5.1 节讨论的弹性驱动轴系统,如图 10.27 所示。很难准确知道谐振频率模型,对谐振频率的估计可高可低。为了研究这个问题,将没有补偿的指针波特图与带有以下两种补偿形式的波特图画在一起:

$$\text{带阻滤波器 \#1:}\ \frac{(s+3+28i)(s+3-28i)}{(s+60)^2}$$

$$\text{带阻滤波器 \#2:}\ \frac{(s+3+34i)(s+3-34i)}{(s+60)^2}$$

在画图以前令每个带阻滤波器增益为 1。

① 参见 R. Jurgen, *Electronic Engine Control Technologies*, SAE International, Troy, MI, 1999 年。

10.5 实际系统中通常存在非线性。研究图 10.48 所示的非线性输入反馈系统。非线性遵循关系式 $u(t)=f(e(t))$,其中,$e(t)$是输入信号,$u(t)$是非线性输出信号。对象为:

$$G_0(s)=\frac{1}{s+1}$$

对于下面的非线性函数,当控制器增益分别为 $k=1,k=10$ 及 $k=100$,且输入为阶跃信号时,求系统稳态响应。

a. $f(e)=0.2(e^3-e)$

b. $f(e)=e+\sin(e) \quad |e|>1$

$\quad =0 \quad |e|\leqslant 1$

c. $f(e)=\tan^{-1}(e)$

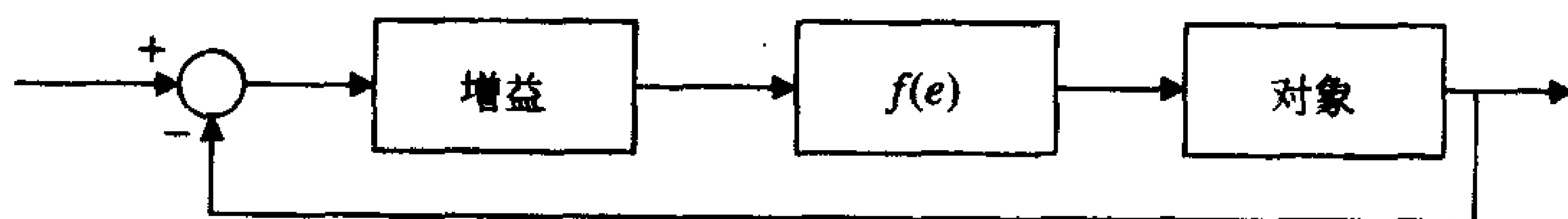


图 10.48 非线性输入的简单系统方框图

10.6 研究图 10.49 所示的导弹,空气的侧向力使导弹相对于其重心旋转,空气产生的力可视为作用于压力中心的一个点力。如果压力中心在重心的前面,则导弹是不稳定的。系统的输入是推力角度 $\Psi(t)$,系统的输出是 $\theta(t)$。空气的阻力可以建模为 $F_d=k_d\sin(\theta)$,其中 k_d 取决于火箭的形状和速度。由火箭发动机产生的外力为 $F_r\sin(\Psi)$。其他相关参数是 l_1 为从火箭的发动机到导弹重心的距离,l_2 为从导弹重心到压力中心的距离,J 是火箭的转动惯量。

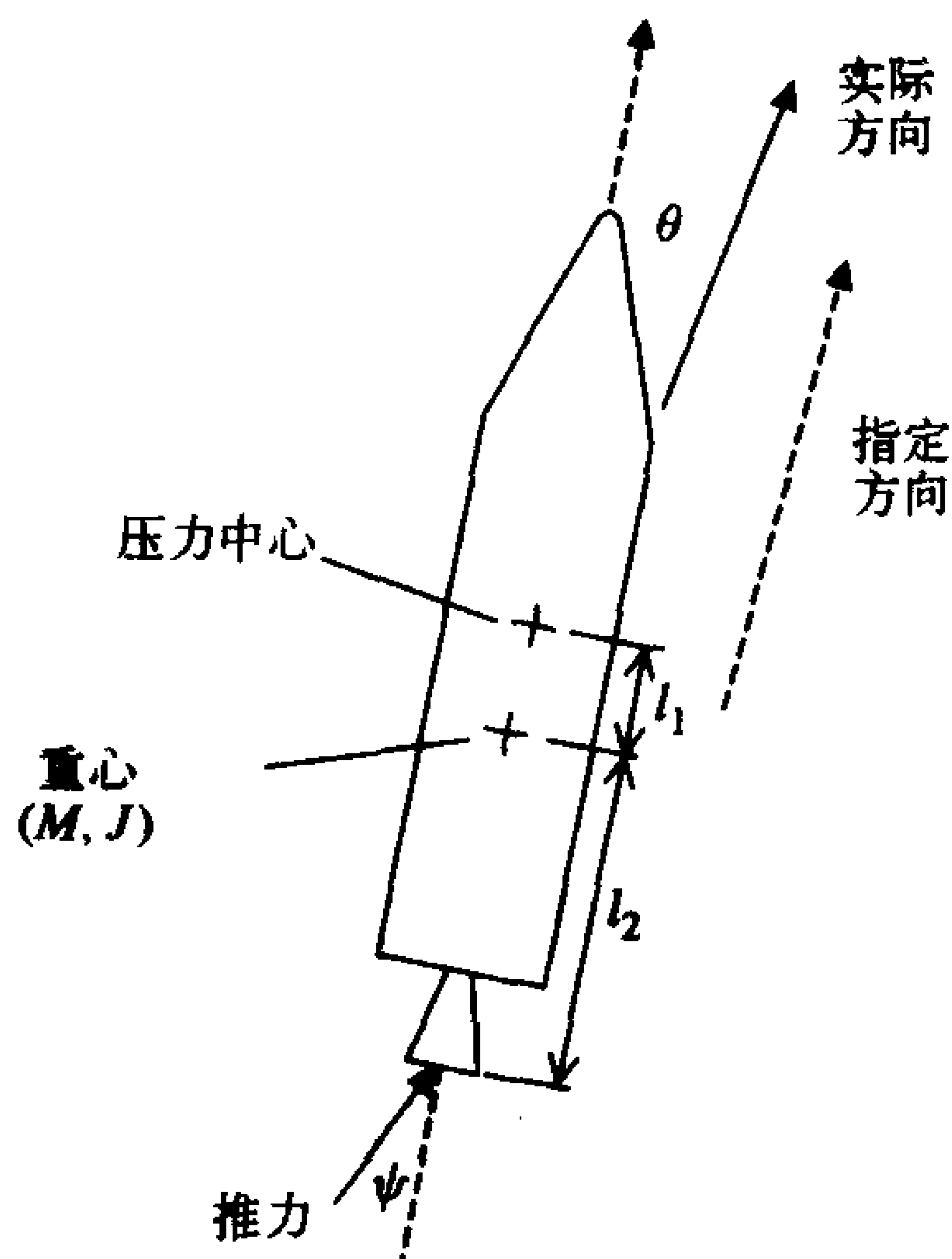


图 10.49 导弹的姿态控制

对于固定的 k_d 和 F_r，当 Ψ 和 θ 很小时，从 Ψ 到 θ 的传递函数为[①]：

$$G(s)=\frac{l_1F_r/J}{s^2-l_2k_d/J}$$

假设 $l_1F_r/J=\ l_2k_d/J=9$ 为关心的工作点。

a. 用超前控制结构：

$$C(s)=k\frac{s+z}{z+p}$$

求使闭环系统稳定的 k,z 及 p 的值。

b. 当火箭的速度变化时，k_d 也变化。在火箭的运行中速度变化是很大的，实际上也改变了 k_d。对(a)中设计的控制器，用根轨迹图确定使系统稳定的 k_d 的范围。

10.7 设计汽车巡行控制器。汽车可以建模为一个具有限制前行阻尼器的物体。从发动机调速汽门角到汽车速度的开环传递函数 $G(s)$ 为：

$$G(s)=\frac{1}{ms+b}$$

$C(s)$ 为 PI 控制器，传递函数为：

$$C(s)=k_p+k_i/s$$

假设汽车的质量 $m=1200$ kg，摩擦系数 $b=70$ Ns/m，路的斜度作为扰动输入。系统的方框图如图 10.50 所示。期望的速度为 $r(t)$，电动机的输入(调速汽门角度)为 $u(t)$，用车速表测得的车速为 $y(t)$。扰动 $d(t)$ 描述了路面斜度的影响程度。

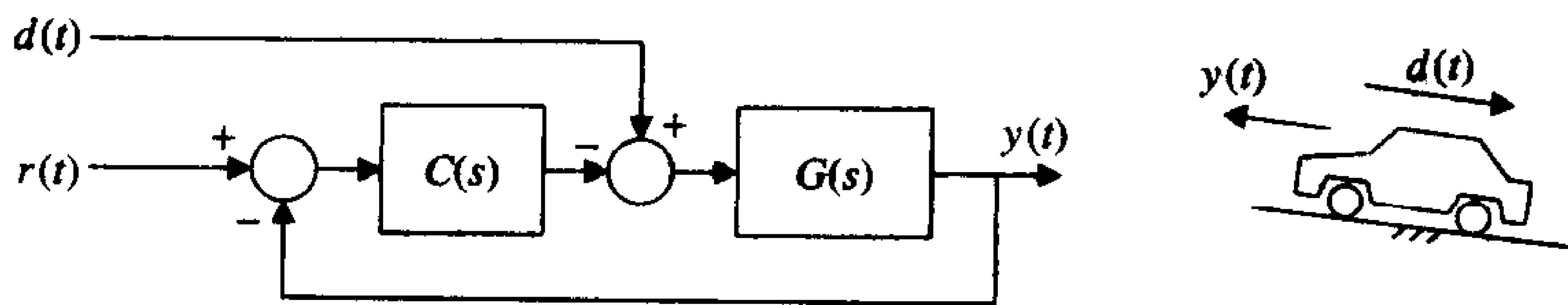


图 10.50 汽车巡行控制方框图

a. 从扰动 $d(t)$ 到速度 $y(t)$ 的传递函数为：

$$G_{cl}(s)=\frac{G(s)}{1+C(s)G(s)}$$

其中，$G(s)$ 为开环响应，$C(s)$ 为前面定义的 PI 传递函数。在同一图上画出 $k_p=100$ 及 $k_i=0$ 时系统开环及闭环的阶跃响应曲线。注意，随着增益的增大扰动在减小。

b. 从速度目标值 $r(t)$ 到速度输出 $y(t)$ 的传递函数为：

$$G_{ry}(s)=\frac{C(s)G(s)}{1+C(s)G(s)}$$

画出 $k_p=100$ 及 $k_i=0$ 的阶跃响应曲线。

c. 比例增益 P 的变化范围在 50 ~ 150 之间，画出(a)和(b)阶跃响应的系统稳态输出。可以看出积分器的作用消除了稳态误差，并且积分器的速度由增益 k_i

① 参见 M. Driels, *Linear Control Systems*, McGraw-Hill, New York, NY, 1996 年。

确定。

10.8 一般的汽车减震系统由弹簧和阻尼器构成,是一个被动系统。为了提高汽车的减震性能,必须采用主动的减震系统。图 10.51 显示了一个汽车车轮的主动减震系统的简化模型,其中 $x(t)$代表路面输入, $y(t)$是车厢的垂直位置。假设轮胎的质量可以忽略,采用速度反馈,$u(t) = C\mathrm{d}x(t)/\mathrm{d}t$。调节器作用在支撑杆与车厢上的力与速度成比例:$C(\mathrm{d}x/\mathrm{d}t - \mathrm{d}y/\mathrm{d}t)$。从 $y(t)$到 $r(t)$的传递函数为:

$$G(s) = \frac{(c+b)s+k}{ms^2+(c+b)s+k}$$

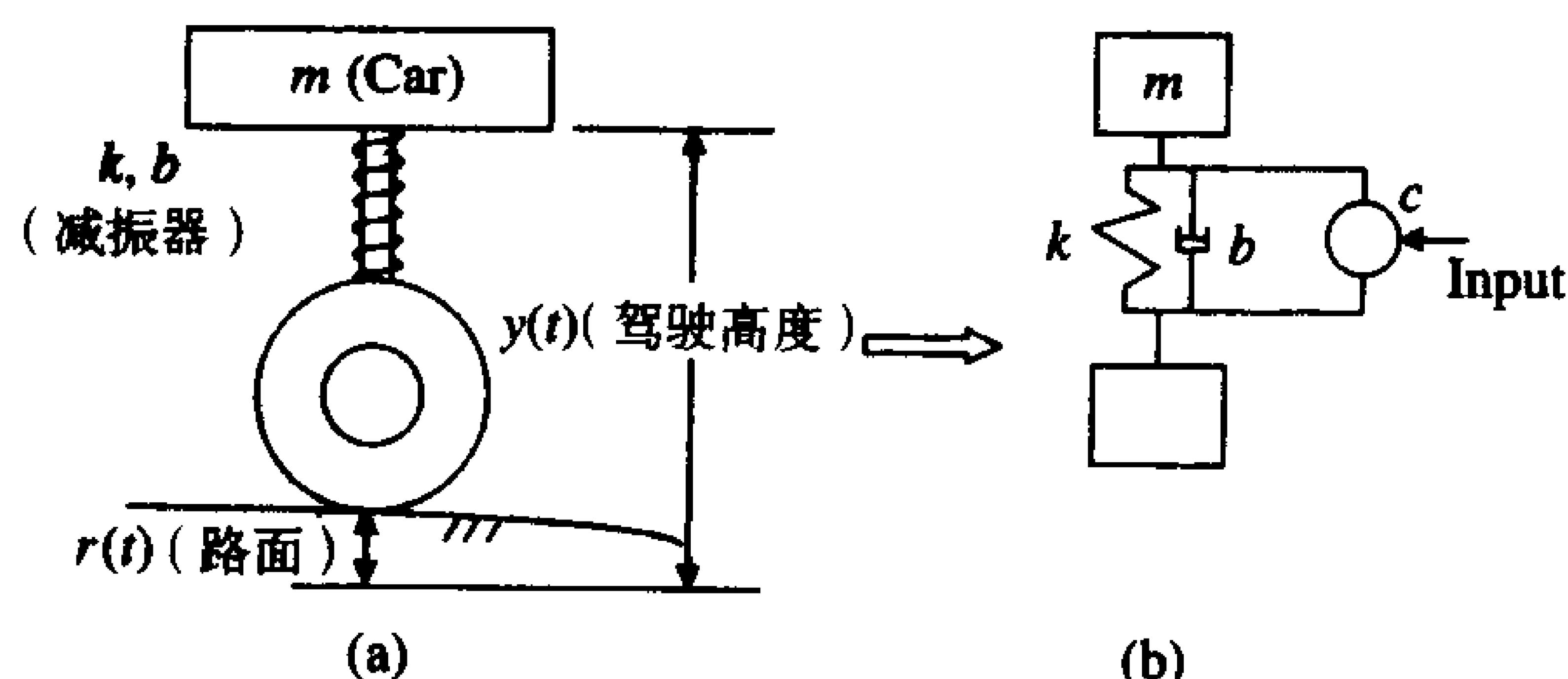


图 10.51 (a) 一个汽车车轮的主动减震模型的简化模型;(b) 质量和弹簧等效系统

a. 用函数 rlocus 和 rlocfind 确定 c 的值,使阻尼系数为 1,即系统处于临界阻尼状态。用函数 rlocus 时注意,分母可以写成 $ms^2+bs+k+cs$。rlocus 的标准形式为:分子[1 0]和分母[m b k]。由 rlocfind 确定的值是 c。将十字线放在闭环极点与实轴首次相交的位置。

b. 分别画出系统带有主动控制及没有主动控制($c=0$)时的阶跃响应和波特图。假设 $m = 5000$ kg, $k = 8\times10^5$ N/m, $b = 12\,000$ Ns/m。

c. 在这个问题中弹簧用 k 计算。采用(b)中的设计,k 的范围在 4×10^5N/m 到 10×10^5 N/m之间,生成阶跃响应的网格图形。比较两个系统的性能。

10.9 为大型慢速行驶的船设计自动驾驶仪,用罗盘的输出作为反馈。控制器对方向舵机构发出命令,命令经过延迟变成期望的位置,从而调整船的运行。对船的Nomoto 方程① 线性化得出下面的方程。没有控制器时操纵系统的开环传递函数为②:

$$G(s) = \frac{s+0.03}{s(s+0.09)(s+0.04)(s-0.0004)}$$

a. 对象有不稳定极点,画出操纵系统的根轨迹。

b. 用超前控制结构:

$$C(s) = k\frac{s+z}{z+p}$$

要求闭环系统响应稳定,超调小于 30%。求 k,z 和p 的值。

c. 若用传感器提供速度信息,则可以使用 PD 控制。因此,$C(s) = k_p + sk_d$,当闭环响应稳定,超调小于 5%,调节时间小于 275 s 时,求 k_p 和 k_d 的值。

---

① 参见 M. Driels,出处同上。

② 参见 C. L. Phillips 和 R. D. Harbor, *Feedback Control Systems*, 第三版, Prentice Hall, Englewood Cliffs, NJ, 1996 年。

10.10 一种带催化转换器的新汽车模型能使废气排放量达标。催化转换器要求严格控制发动机的空气/燃料比(A/F)、点火时间和废气再循环。研究空气/燃料比调节任务,以 A/F 作为输出,炭化器的传递函数如下[①]:

$$G(s)=\frac{4e^{-T_d s}}{s+4}$$

其中,时间延迟 T_d 为 0.2 s,函数 pade 求取 T_d 的近似值,也可以将传递函数的输出延迟设为 0.2。因为函数 pade 是 MATLAB 中专门用于时间延迟计算的,所以使用起来很方便。

a. 假设控制器设计中忽略时间延迟,令控制器为 PI 控制,可写为:

$$C(s)=k_p+\frac{k_i}{s}$$

令 $k_i=2$ 并选择 k_p 的值,使得单位阶跃响应的上升时间小于 0.4 s。确定系统的阶跃响应。

b. 研究将时间延迟信号馈送到控制器的过程,如图 10.52 所示。控制器中的附加补偿单元包括对象模型及其时延,称为史密斯(Smith)预估器。采用超前控制机构

$$C(s)=k\frac{s+z}{z+p}$$

要求闭环系统稳定且没有超调,上升时间小于 0.2 s,求 k, z 和 p 的值。将结果与(a)中的 PI 控制器进行比较。

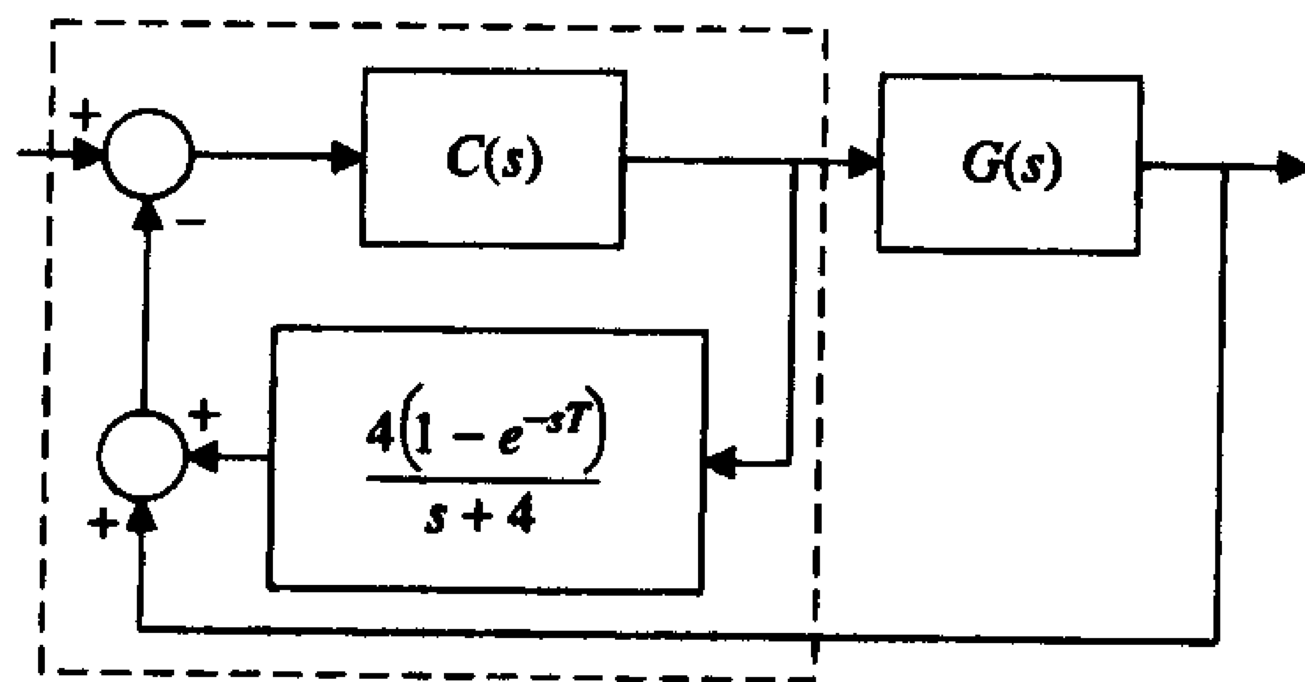


图 10.52 带有史密斯预估器的废气控制系统

c. 假设时间延迟和对象都没有精确模型,确定系统延迟时间为 0.3 s 和 0.1 s 的阶跃响应。控制器采用(b)中所设计的控制器,对象模型的直流增益为 1.2,极点不是 -4 而是 -5。从结果看,对延迟时间估计是高点好还是低点好?

10.11 研究图 10.53 所示电动自行车方框图。倾斜仪是用来检测车体偏离垂直方向角度 $y(t)$的。倾斜仪的输出与期望垂直角度 $r(t)$相比较,将误差送回控制器并产生操纵信号。系统的任何扰动 $d(t)$都作为电动自行车的输入。方框图 10.53 的传递函数为:

$$G(s)=\frac{9}{s^2+9}$$

① 参见 B. Kuo, *Automatic Control Systems*,第七版,Prentice Hall, Englewood Cliffs, NJ, 1995 年,第 815 页。

$$F(s)=\frac{\omega^2}{s^2+2\xi\omega s+\omega^2}$$

$$C(s)=k\frac{s+z}{s+p}$$

其中,$C(s)$为超前控制器。

a. 微型机械倾斜仪的调节时间为 0.2 s,带宽为 125 Hz。传感器的参数 $\omega=250\pi$,$\xi=20/\omega$。当单位阶跃响应的超调小于 20%,调节时间小于 4 s 时,求 k,z 和 p 的值。

b. 假设有另一种类型的倾斜仪可以选择。它基于钟摆的原理,谐振频率是 7.4 Hz ($\omega=14.8\pi$),阻尼系数 $\xi=0.4$。这样的倾斜仪可以使用吗?

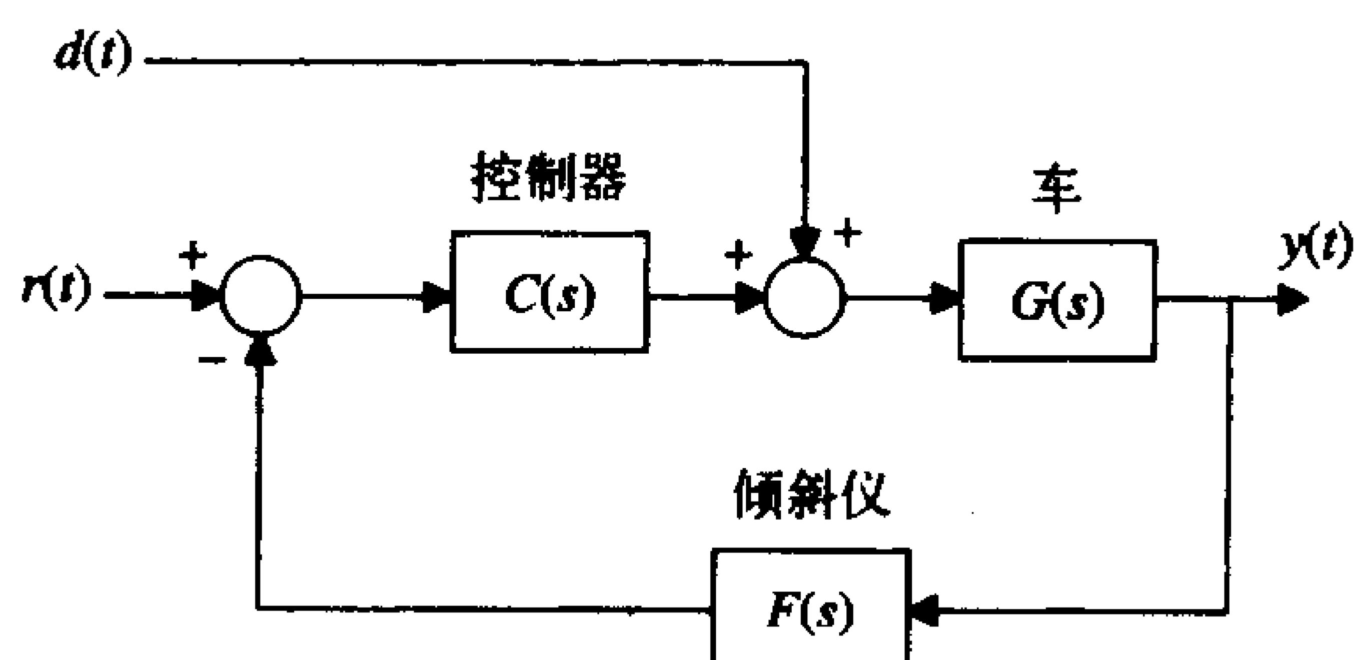


图 10.53 电动自行车的方框图

参考文献

1. D. K. Anand and R. B. Zmood, *Introduction to Control Systems*, Butterworth and Heinmann, Ltd., Oxford, England, 1995.
2. E. Chowanietz, *Automobile Electronics*, SAE International, Troy, MI, 1995.
3. R. Dorf and R. Bishop, *Modern Control Systems*, Addison-Wesley, Reading, MA, 1997.
4. M. Driels, *Linear Control System Engineering*, McGraw-Hill, New York, NY, 1996.
5. G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed., Addison-Wesley, Reading, MA, 1994.
6. B. Friedland, *Advanced Control System Design*, Prentice Hall, Englewood Cliffs, NJ, 1996.
7. R. Jurgen, *Electronic Engine Control Technologies*, SAE International, Troy, MI, 1999.
8. B. Kuo, *Automatic Control Systems*, Prentice Hall, Englewood Cliffs, NJ, 1995.
9. W. Levine, *The Control Handbook*, CRC Press, Boca Raton, FL, 1996.
10. L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Upper Saddle River, NJ, 1999.
11. N. Nise, *Control Systems Engineering*, Addison-Wesley, Reading, MA, 1995.
12. U. Ozguner, H. Goktas, and H. Chan, "Automotive Suspension Control Through a Computer Communication Network," *Proceedings of First IEEE Conference on Control Application*, Vol. 2, 1992, pp, 895 - 900.
13. Phillips and R. Harbor, *Feedback Control Systems*, Prentice Hall, Englewood Cliffs, NJ, 1996.

第11章 流体力学

本章对流体力学和空气动力学中的几类问题加以分析,并用不同可视化技术表示几种流场。

11.1 流体静力学

在流体静力学中,当高度一定时,任意平面上均质液体的静压强是恒定不变的。高度一旦变化则压强也随之而变化。满足以下方程:

$$\frac{\mathrm{d}P}{\mathrm{d}z} = -\rho g \tag{11.1}$$

式中,$\rho = \rho(z)$是密度分布函数,g 是重力加速度(9.81 m/s^2),P 是压强,z 是笛卡儿坐标的垂直分量,向上为正。利用式(11.1)解决两个流体静力学问题。

11.1.1 标准大气压下的压强分布

如果空气为理想气体,则有:

$$\rho = P/(RT)$$

式中,$R = 287.13$ J/(kg ℃K)是理想气体常数,T 是热力学温度,单位为开尔文。在这种条件下,结合式(11.1),可以得到:

$$P(z) = P_0 \exp\left[-\frac{g}{R}\int_0^z \frac{\mathrm{d}z}{T}\right] \tag{11.2}$$

式中,在 $z = 0$ 处的压强 $P_0 = 101\ 330$ Pa。假定在标准大气压下温度分布呈分段线性,这种线性分布可由不同海拔高度下的一组温度值定义,表11.1体现了这一规律。

表11.1 标准大气压下的温度与海拔高度的依变关系

海拔高度(m)	温度(℃)
0.0	15.0
11 000.0	-56.5
20 100.0	-56.5
32 200.0	-44.5
47 300.0	-2.5

例11.1 温度和压强随海拔高度的依变关系

虽然式(11.2)很容易求解,但是必须对数值进行估测。以下程序对式(11.2)求解,并画出 P 和 T 随 z 变化的函数曲线,用函数 interp1 实现表11.1中温度数值的线性插值,这里积分下限取0.1 m。结果如图11.1所示。

```
gravity = 9.81; p0 = 101330; R = 287.13;
tempC = [15, -56.5, -56.5, -44.5, -2.5];
```

```
z=[0,11000,20100,32200,47300];
invertemp=1./(tempC+273.15);
np=18;
goverR=gravity/R;
elevation=linspace(0,z(end),np);
pressure=zeros(1,np);
intarg=inline('interp1(z,invertemp,elevation)','elevation','z','invertemp');
for i=1:np
  pressure(i)=p0*exp(-goverR*quadl(intarg,0.1,elevation(i),[],[],z,…inversetemp));
end
subplot(1,2,1)
plot(tempC,z/1000.0,'k-s')
axis([-60 20 0 elevation(end)/1000.0])
xlabel('Temperature(C)')
ylabel('Elevation (km)')
subplot(1,2,2)
plot(pressure/1000.0,elevation/1000.0,'k')
axis([0,110,0,elevation(end)/1000.0])
ylabel('Elevation (km)')
xlabel('Pressure(kPa)')
```

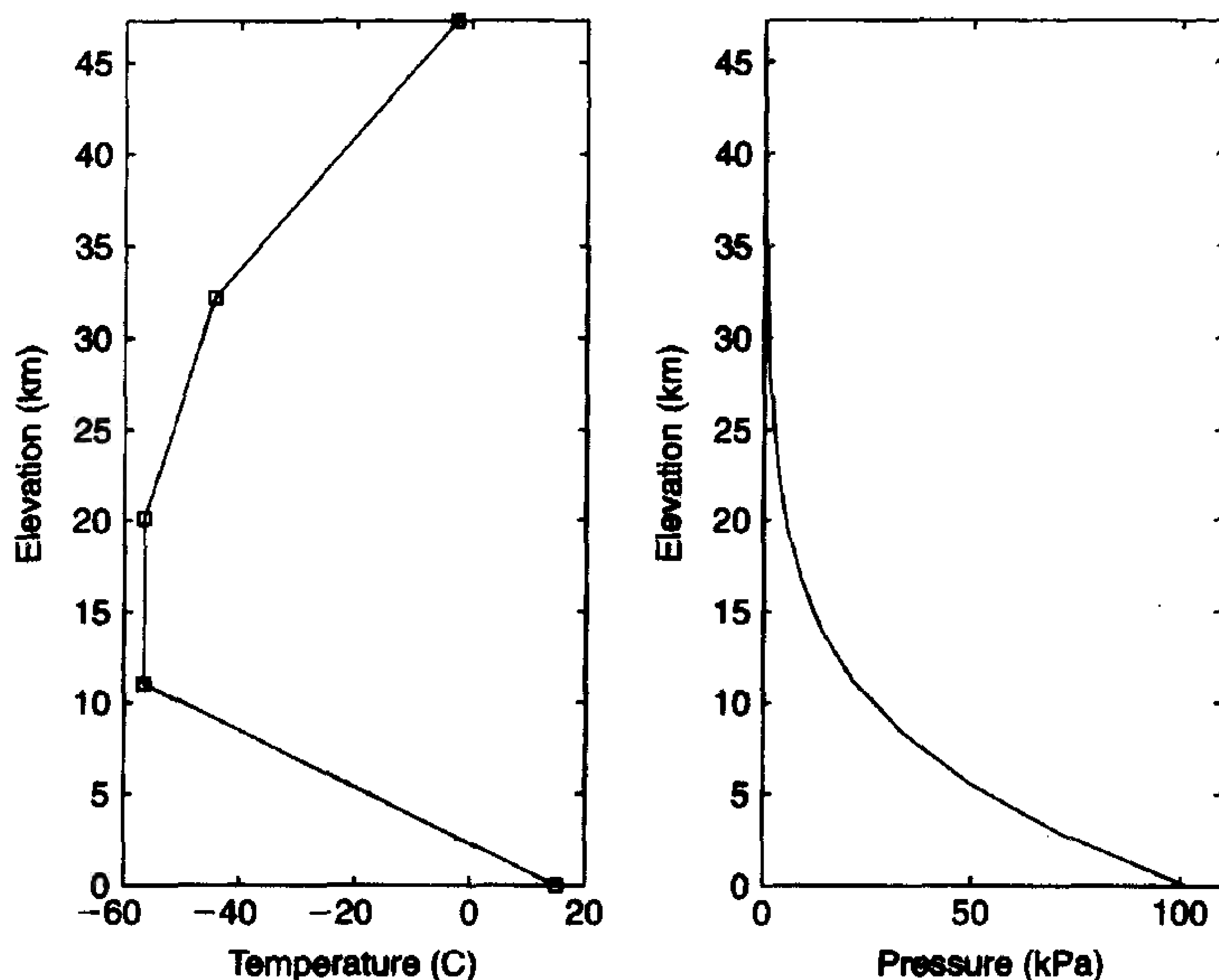


图 11.1 标准大气压:(a)表 11.1 中温度与海拔高度的依变关系;(b)式(11.2)中的压强与海拔高度的依变关系

11.1.2 平面闸门的作用力

研究图 11.2 所示的水库问题。水库的一个壁是一个可倾斜的金属闸门,可以绕底部转动,其重量为 W,长度为 L。水库在垂直于书页表面方向上的宽度为 B。初始时闸门垂直放置,水位刚好达到闸门的顶部,水的体积为 $V_w = aLB$。闸门由一根杆支撑,沿杆方向作用在闸门上的力为 F_{rod}。

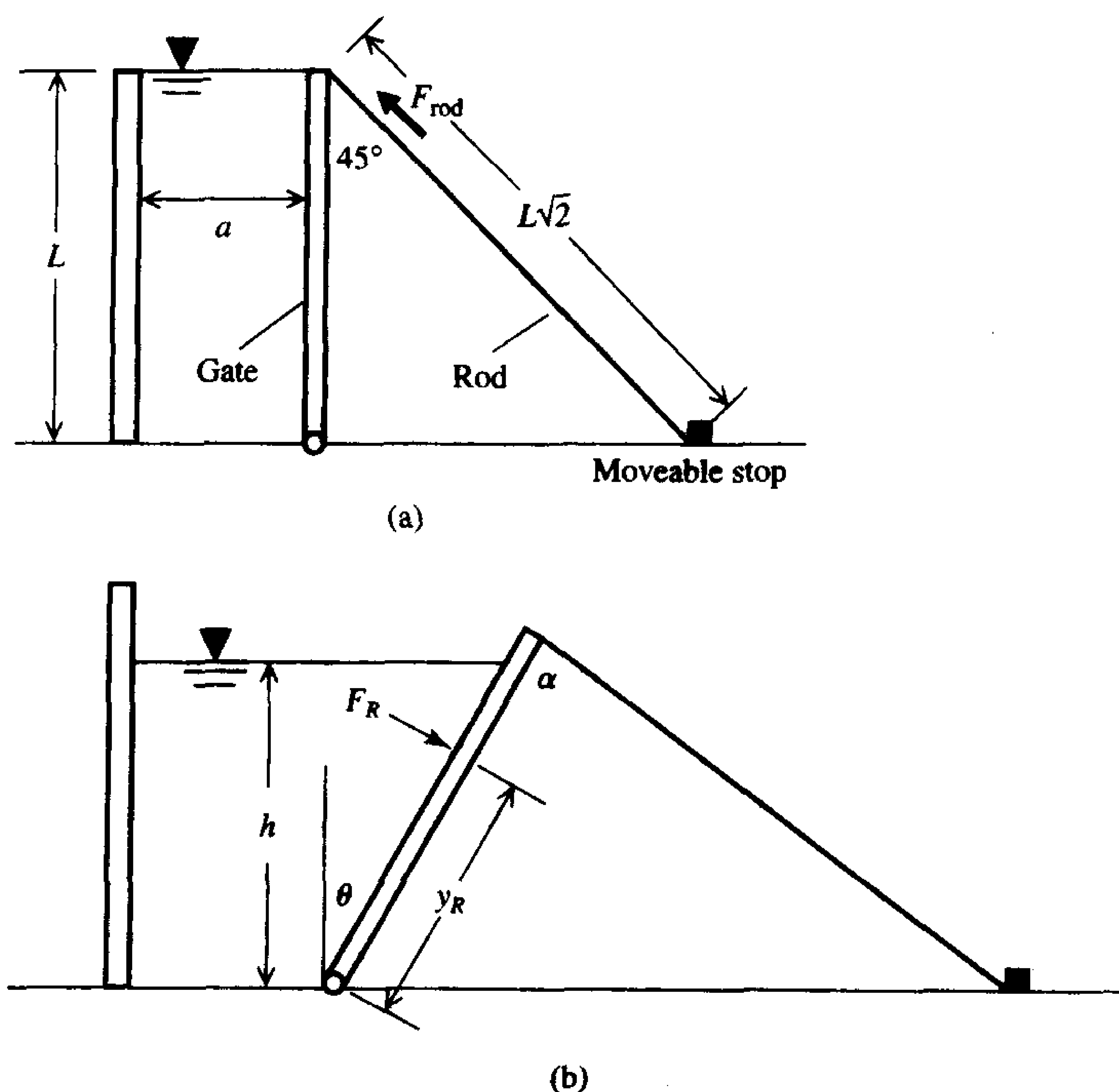


图 11.2 具有可倾斜门的水库:(a)竖直的门;(b)打开一定角度 θ 的门

杆的另一端被一障碍物支撑住,这个障碍物可以向右移动,使得闸门能绕转动轴顺时针方向旋转。由于闸门的倾角 θ 只能小于或等于某个临界角度 θ_{max},所以水位只能低于或等于闸门的高度。当 $\theta > \theta_{max}$ 时,水会溢出闸门顶部。

由水库底部、固定的壁、闸门和闸门顶部的平面围成的体积为:

$$\frac{V}{V_w} = \cos\theta + \frac{L}{2a}\cos\theta\sin\theta \tag{11.3}$$

当满足以下条件时,水将溢出水坝:

$$\frac{V}{V_w} < 1.0$$

根据任意角度 θ 下的水的体积与 V_w 的等式关系可以得到水位高度 h 对 θ 的关系式,即:

$$\frac{V_w}{B} = aL = ah + 0.5h^2\tan\theta \tag{11.4}$$

通过对转动轴力矩的测量可以求得 F_{rod} 的值,即:

$$F_{rod} = \frac{F_R y_R + 0.5WL\sin\theta}{L\sin\alpha} \tag{11.5}$$

其中,F_R 是水对闸门的总作用力,y_R 是转动轴到压力中心的距离,角度 α 如图 11.2(b)所示,且由以下方程式给出:

$$\alpha = \theta + \cos^{-1}(\cos\theta/\sqrt{2})$$

从流体静力学的平衡关系式中可以得出:

$$F_R = \frac{B\rho g h^2}{2\cos\theta}$$

$$y_R = \frac{h}{2\cos\theta} - \frac{2I_{xx}\cos^2\theta}{Bh^2}$$

式中,

$$I_{xx} = \frac{Bh^3}{12\cos^3\theta}$$

是闸门浸在液体中轴的惯性矩。则式(11.5)变为:

$$F_{rod} = \frac{1}{L\sin\alpha}\left[\frac{B\rho g h^3}{6\cos^2\theta} + \frac{WL}{2}\sin\theta\right] \tag{11.6}$$

以下将给出这些结果的说明。

例 11.2 水库特性

在以下的程序代码中将求出 θ_{max} 值,并在 $0 \leqslant \theta \leqslant \theta_{max}$ 范围内绘出水深 h 和 F_{rod} 的曲线,结果如图 11.3 所示。假定 L = 10 m, a = 5 m, B = 10 m, W = 100 000 N。此外,V/V_w 由式(11.3)计算得出,可以绘出它对 θ 的关系曲线。MATLAB 中的 fzero 函数可以用来求 θ_{max},θ_{max} 位置标在图 11.3(a)上。在计算 θ_{max} 的过程中,inline 函数创建 **MaxTheta**,其返回值为 $1 - V/V_w$。为了求得 h 对 θ 的关系曲线,用 roots 求解二次方程(11.4)中 h 的正根。水库盛水最多时确定 θ 的值也很有意义,可以由 fminbnd 调用 **MaxTheta** 来实现。图 11.3 中显示了 θ = 30°时的计算结果。求解这些值并显示这些结果(包括图 11.4 中 F_{rod} 对 θ 的关系曲线)的程序代码为:

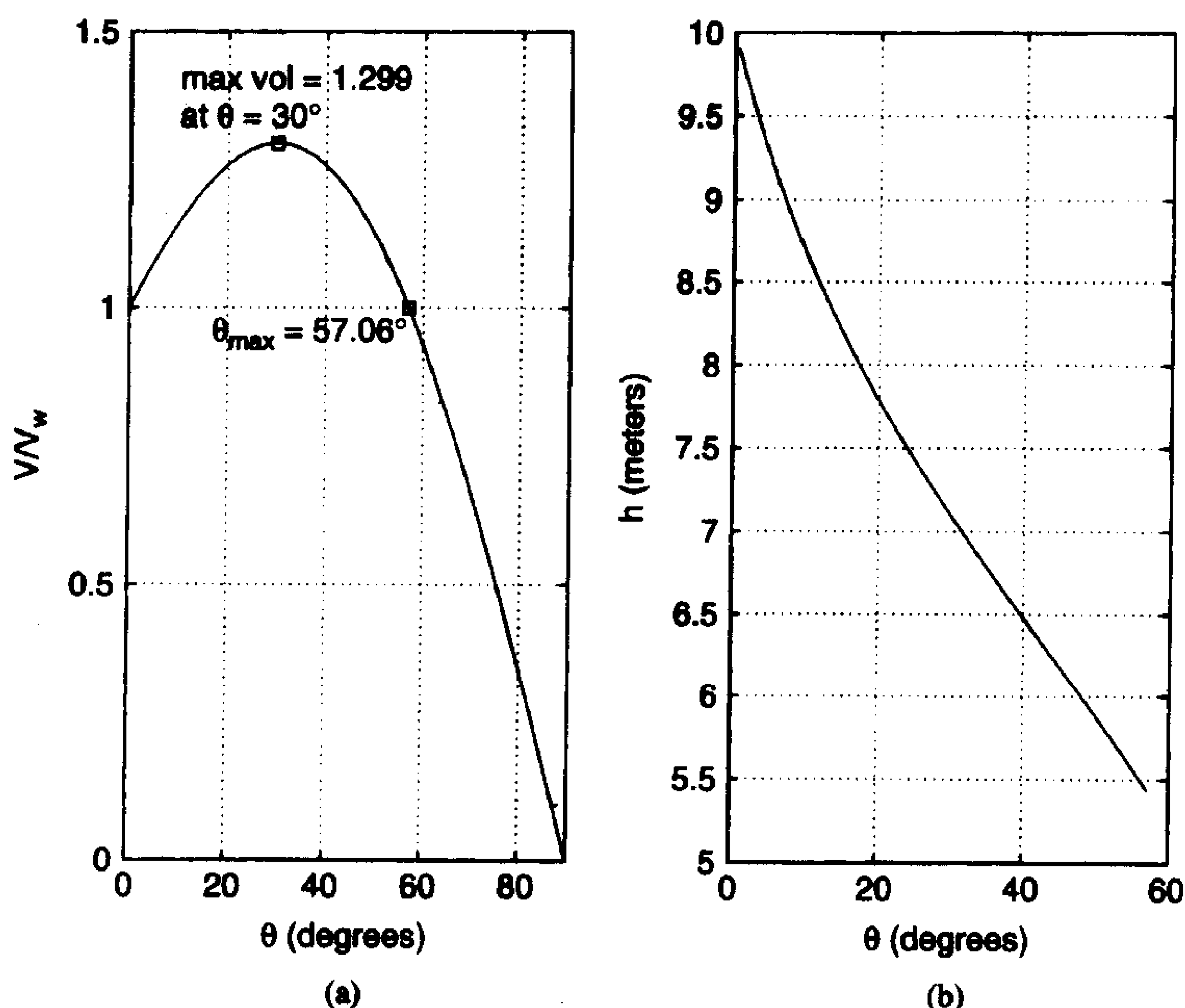


图 11.3 可转动门构造的结果图:(a) V/V_w 与 θ 的对应关系;(b) h 与 θ 的对应关系

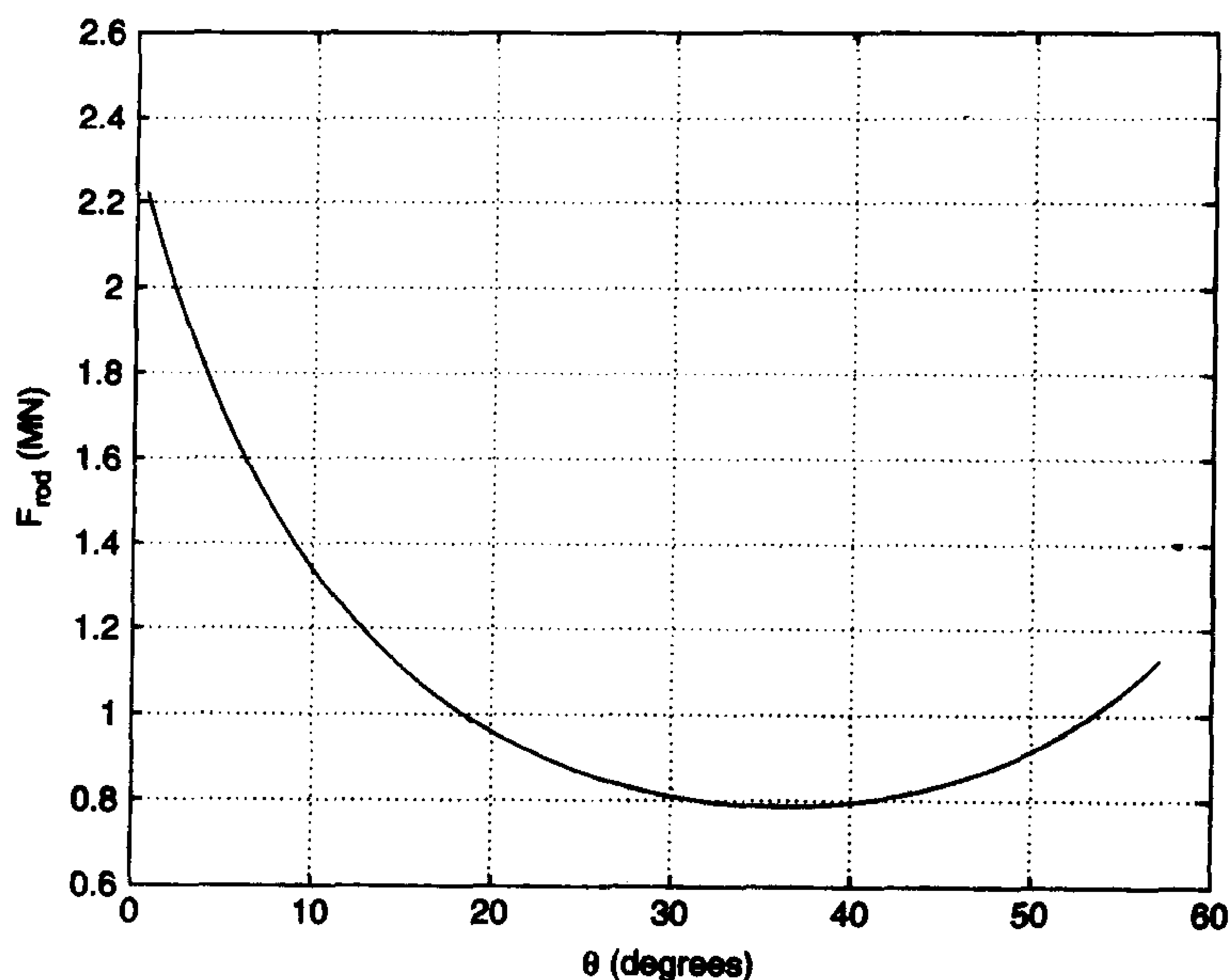


图 11.4 使门关闭的力与 θ 的对应关系

```
a = 5.0; L = 10.0; B = 10.0; rho = 1000.0;
grav = 9.81; W = 100000.0; ratio = L/a;
theta = linspace(0.0,pi/2.0);
VoverVw = cos(theta) + 0.5 * ratio * cos(theta). * sin(theta);
options = optimset('display','off');
figure(1)
subplot(1,2,1)
plot(theta * 180/pi, VoverVw,'k')
axis([0.0,90.0,0.0,1.5])
ylabel('V/V _ w')
xlabel(' \ theta(degrees)')
grid on
hold on
MaxTheta = inline('1 - (cos(theta) + 0.5 * ratio * cos(theta). * sin(theta))','theta',…'ratio');
ThetaMaxDeg = fzero(MaxTheta,[0.01,pi/2.0],options,ratio) * 180/pi;
plot(ThetaMaxDge,1.0,'sk')
text(19,0.95,[' \ theta _{max} = ',num2str(ThetaMaxDeg,4)' \ circ'])
ThetaMaxVol = fminbnd(MaxTheta,0.0,ThetaMaxDeg * pi/180,options,ratio);
MaxVol = 1 - MaxTheta(ThetaMaxVol,ratio);
plot(ThetaMaxVol * 180/pi,MaxVol,'ks')
text(10,MaxVol + 0.1,['max vol = 'num2str(MaxVol,4)])
text(10,MaxVol + 0.05,['at \ theta = 'num2str(ThetaMaxVol * 180/pi,4)' \ circ'])
subplot(1,2,2)
theta = linspace(0.01,ThetaMaxDeg * pi/180);
h = zeros(1,length(theta));
for i = 1:length(theta)
  = roots([tan(theta(i)) * 0.5,a, - a * L]);
  rh(i) = r(2);
end
plot(theta * 180.0/pi,h,'k - ')
ylabel('h(meters)')
```

```
xlabel('\theta(degrees)')
grid on
figure(2)
Frod = ((B * rho * grav * h.^3)./(6 * cos(theta).^2) + 0.5 * W * L * sin(theta))./...
        (L * sin(theta + acos(cos(theta)/sqrt(2))));
plot(theta * 180.0/pi,Frod * 1e - 6,'k-')
ylabel('F_{rod}(MN)')
xlabel('\theta(degrees)')
grid on
```

11.2 内部黏性流动

在管道中的层流黏性流动和紊流黏性流动是一大类问题。以下给出了几种问题的解决方法。管道中低雷诺数的流动为层流,PDE 工具箱可以计算流场及沿管道而降低的压强;高雷诺数的流动为紊流,流动和压强的势差都可以用科尔布鲁克方程式求出[①]。

11.2.1 不同截面水平管道的层流

利用 MATLAB PDE 工具箱,可以较直接地计算出各种形状横截面的水平管道中充分发展的层流场。在大部分教科书中对圆管的分析使用了这种方法[②]。因此,首先求解这个流场,再将计算出的流速最大值和平均值与解析法求解结果相比较。假定管道半径 R 为 5.0 mm,流体的动力黏性系数 μ 为 0.38 N·s/m²,轴向压强梯度 dP/dz 为 1.0×10^6 Pa/m。

轴向流场的微分方程为:

$$\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=\frac{1}{\mu}\frac{\mathrm{d}P}{\mathrm{d}x} \tag{11.7}$$

式中,$u(y,\ z)$是 x 轴方向上的速度,y 和 z 是管道横截面的笛卡儿坐标,点(0, 0)是管道的中心点,管道壁的坐标$(y,\ z)$满足:

$$y^2+z^2=R^2$$

壁面处滞流的边界条件为 $u=0$。

下面用 pdetool 来求解这个问题[③]。在 pdetool 中首先创建一个以点(0, 0)为中心,以 1.0 为半径的圆,这就意味着创建了一无量纲坐标系 $y'=y/R$ 和 $z'=z/R$。式(11.7)变为:

$$\frac{\partial^2 u}{\partial y'^2}+\frac{\partial^2 u}{\partial z'^2}=\frac{R^2}{\mu}\frac{\mathrm{d}P}{\mathrm{d}x}=f$$

在 pdetool 中,首先进入 *Boundary* 菜单选择 *Boundary Mode*,在圆的四条弧线上确定第立克特边界条件($u=0$),然后选定椭圆偏微分方程,参数为 $c=1,\ a=0,\ f=65.79$ m/s $((0.005^2/0.38)\times10^6)$。其次,初始化网格并进行两次改进,结果以等高线图形式绘出,如图 11.5 所示。

为了从求解过程中得到其他的定量数据,将求解结果 u 和网格描述的量 p,e,t 一同输出

---

① 参见 C.E.Colebrook,"Turbulent Flow in Pipes with Particular Reference to the Transition Region Between Smooth and Rough Pipe Laws,"*Journal of the Institute of Civil Engineers*, London, 第 11 卷, 1939 年, 第 133 ~ 156 页。

② 参见 B.R.Munson, D.F.Young 和 T.H.Okiishi, *Fundamental of Fluid Mechanics*, John Wiley & Sons, New York, NY,1998 年。

③ 有关如何使用 pdetool 的详细内容参见 8.7 节。

到 MATLAB 命令窗口上，速度的最大值出现在管道横截面的中心点处。为求得该值，可在 MATLAB 命令窗口中输入语句：

```
umax = max(u)
```

结果为 $umax = 16.435$，非常接近理论数值 16.447。

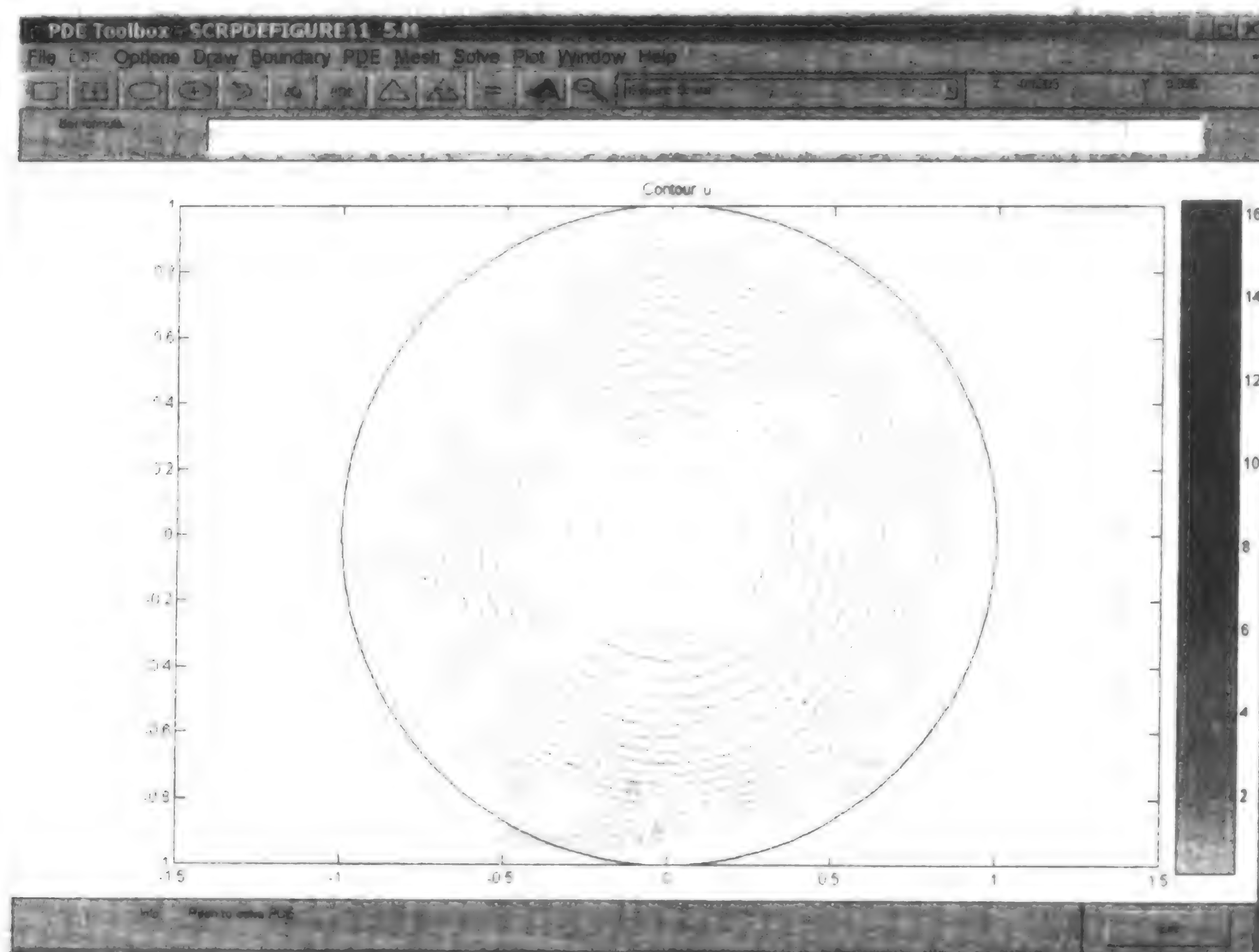


图 11.5 圆管中的轴向流场。最内层等高线的数值为 16

计算流量 Q 的公式为：

$$Q = \iint_A u \mathrm{d}A \approx \sum_{i=1}^{N_t} u_i \Delta A_i$$

式中，A 是管道的横截面积，N_t 是网格中三角形的数量，u_i 为每个格状三角形中心处的速度，ΔA_i 是每个格状三角形的面积。在以下的程序代码中，用 pdeintrp 函数确定 u_i，用 pdetrg 函数确定 ΔA_i：

```
ui = pdeintrp(p,t,u);
DeltaAi = pdetrg(p,t);
Q = sum(DeltaAi. * ui)
```

将 u, p, e 和 t 输出到 MATLAB 窗口后，执行以上程序得到 $Q = 25.806$，平均速度 $u_{ave} = Q/A = 25.806/\pi = 8.214$，非常接近最大速度理论值的一半，即 $16.447/2 = 8.224$。对网格做进一步改进会使计算值和理论值更加一致。

也可以绘出速度与管道直径的函数图形：$y' = 0, -1 \leqslant z' \leqslant 1$。为此，首先必须用 tri2grid 函数将三角网格数据添加到位于管道横截面半径的点上。结果如图 11.6 所示。

图中曲线具有理论方法得到的抛物线型轮廓。其实现代码为：

```
z = linspace( - 1,1,25);
uyz = tri2grid(p,t,u,0,z);
```

```
plot(uyz,z)
axis([0,1800,-1.0,1.0])
ylabel('z')
xlabel('Axial velocity')
```

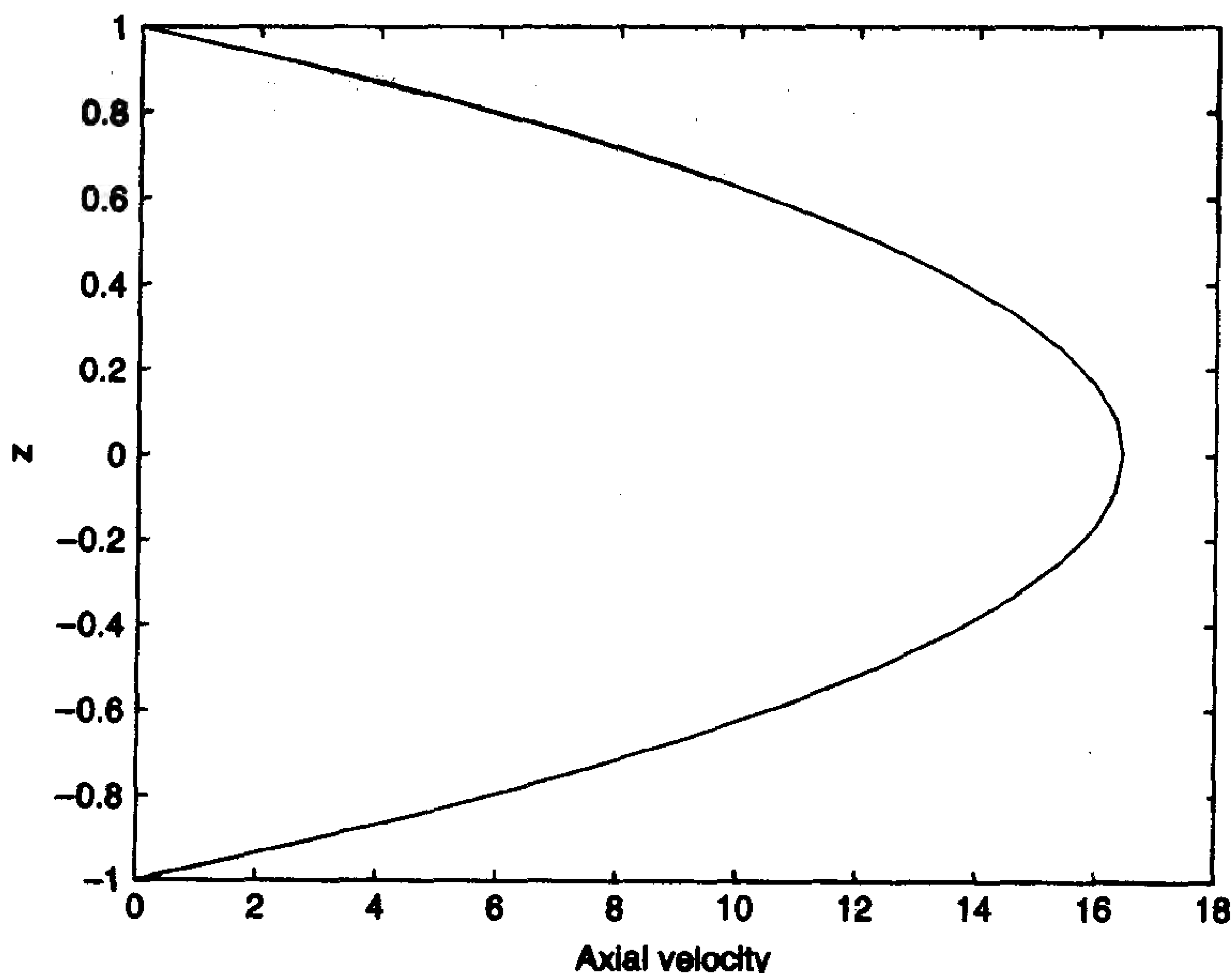


图 11.6 管道内的轴向流速与半径位置的关系

11.2.2 垂直管道中向下的流动

一个立于垂直方向上的光滑管道,长度为 L,直径为 D。管道中通过密度为 ρ,运动黏度为 v 向下流的水(如图 11.7 所示)。对于流动速率来说,重力引起的压强升高与流动引起的压强降低相互抵消,即管道中的静态压强与管道长度无关。

水头损失公式为:

$$\frac{P_1}{\rho g}+\frac{V_1^2}{2g}+z_1=\frac{P_2}{\rho g}+\frac{V_2^2}{2g}+z_2+\frac{\lambda L V^2}{2gD} \tag{11.8}$$

式中,P 是压强,V 是平均流动速度,z 是高度,λ 是摩擦系数。该问题中,$P_1=P_2$,$V_1=V_2$,$z_1-z_2=L$;因此,式(11.8)可简化为:

$$\lambda=\frac{2gD^3}{v^2 R_e^2} \tag{11.9}$$

式中,R_e 是雷诺数,可定义为:

$$R_e=\frac{VD}{v} \tag{11.10}$$

不同粗糙度(k/D)管道的 λ 与 R_e 的科尔布鲁克关系式为(参见习题 5.19):

$$\frac{1}{\sqrt{\lambda}}=-2\log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}}+\frac{k/D}{3.7}\right) \quad R_e\geqslant 4000 \tag{11.11}$$

该例中,$k=0$。消去式(11.9)和式(11.11)中的 λ,可以得到管流的雷诺数和期望流动速率的

一个经验公式。

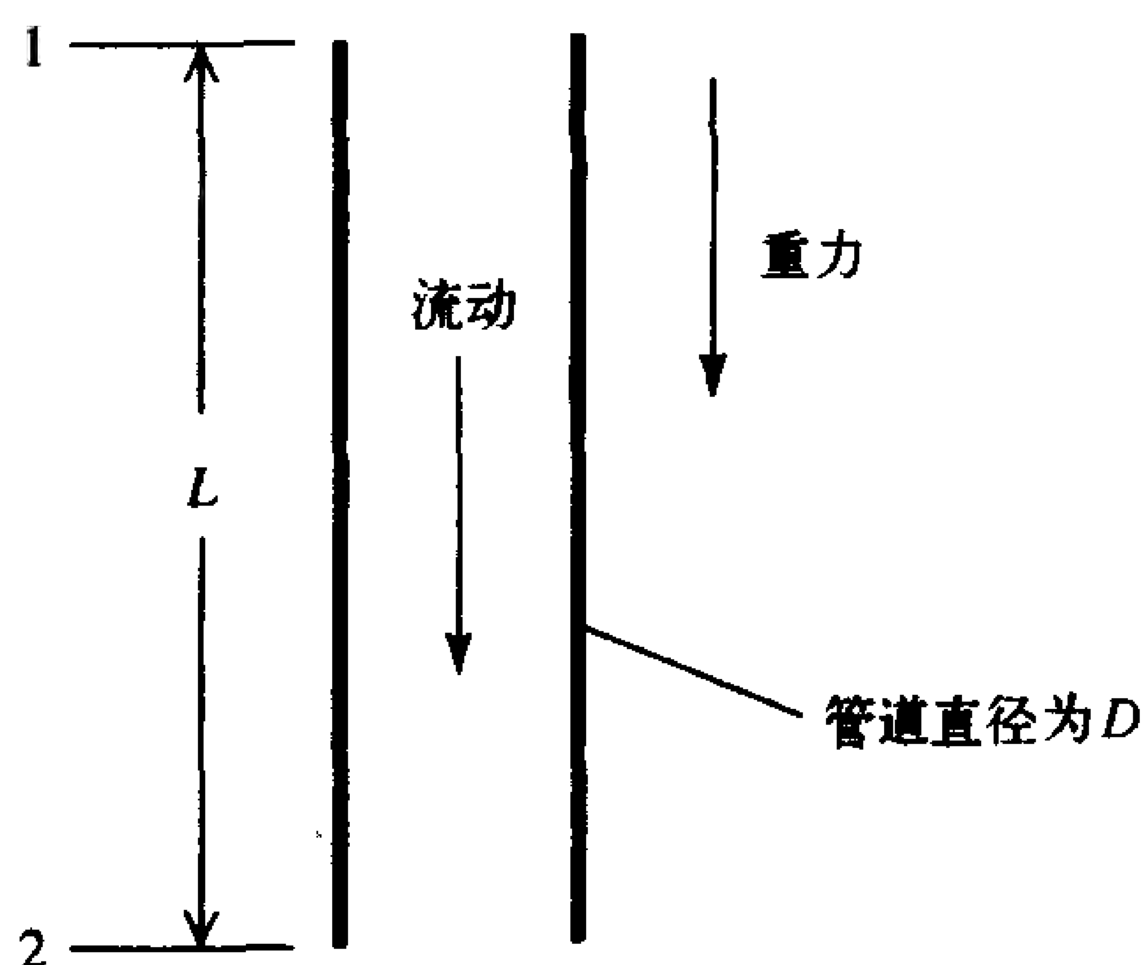


图 11.7 直径为 D 的垂直管道内的流体

以下将用一个例子说明这些结果。

例 11.3 管内的流动速率

研究一个立于垂直方向上的光滑管道，直径 $D = 4.0$ cm。管道中通过密度为 $\rho = 1000.0$ Kg/m^3，运动黏度 $v = 1.2 \times 10^{-6}$m^2/s 向下流的水，求其流动速率。由 **Colebrook-Friction** 函数估算科尔布鲁克方程表达式，程序代码如下：

```
function Colebrook
diameter = 0.04; gravity = 9.81;
nu = 1.2e-6; kOverD = 0.000;
options = optimset('display','off');
Re = fzero(@ColebrookFriction,[1e3,1e7],options,nu,kOverD,gravity,...diameter;)
disp(['Re = ',num2str(Re)])

disp(['Flow Rate = 'num2str(pi * diameter * Re * nu/4)'m^3/s'])
function value = ColebrookFriction(Re,nu,kOverD,gravity,diameter)
lambda = 2 * gravity * diameter^3/(nu * Re)^2;
value = 1/sqrt(lambda) + 2 * log10(kOverD/3.7 + 2.51/(Re * sqrt(lambda)));
```

执行结果为：

```
Re = 240405.8408
Flow Rate = 0.0090631 m^3/s
```

流量为 $Q = \pi D^2 V/4$。

11.2.3 三水库问题

三水库问题①是指三个不同海拔高度的水库连接到同一个交叉点 J 处(如图 11.8 所示)。若给定管道的下列参数：长度 L_j，直径 d_j，管道的粗糙度 k_j 和各个水库的高度 h_j，则可确定每个管道的相应流速 Q_j 和流向。方法如下，如果在交叉点 J 处安装一个开口的管子，则管子中的水可以升到某个未知高度 h_p，P 点和 J 点的高度差就是交叉点处的压差。其次，在 J 处每个管道的总体流量必为 0，即：

① 参见 N.H.C.Hwang 和 C.E.Hita，*Fundamentals of Hydraulic Engineering Systems*，第二版，PrenticeHall，Englewood Cliffs，NJ，1987 年，第 106 ~ 110 页。

$$\sum_{j=1}^{3} Q_j = 0 \tag{11.12}$$

流量 Q_j 为正值,表示水流向交叉点,Q_j 为负值,表示从交叉点流出。

各管道的流量由下式得出:

$$Q_j = 0.25\pi d_j^2 V_j s_j \quad j = 1,2,3$$

上式中,

$$V_j = s_j \sqrt{\frac{2gd_j \mid \Delta h_j \mid}{\lambda_j L_j}} \quad \Delta h_j = h_j - h_p \quad j = 1,2,3$$

s_j 表示 Δh_j 的符号,重力加速度 $g = 9.81\ \text{m/s}^2$, 管道摩擦系数 λ_j 由式(11.11)得出,是 R_{ej} 的函数,R_{ej} 表达式为[参见式(11.10)]:

$$R_{ej} = \frac{V_j d_j}{v} \quad j = 1,2,3$$

式中,20℃时水的运动黏度 $v = 1.002 \times 10^{-6}\ \text{m}^2/\text{s}$ 确定式(11.12)的 h_p 值。

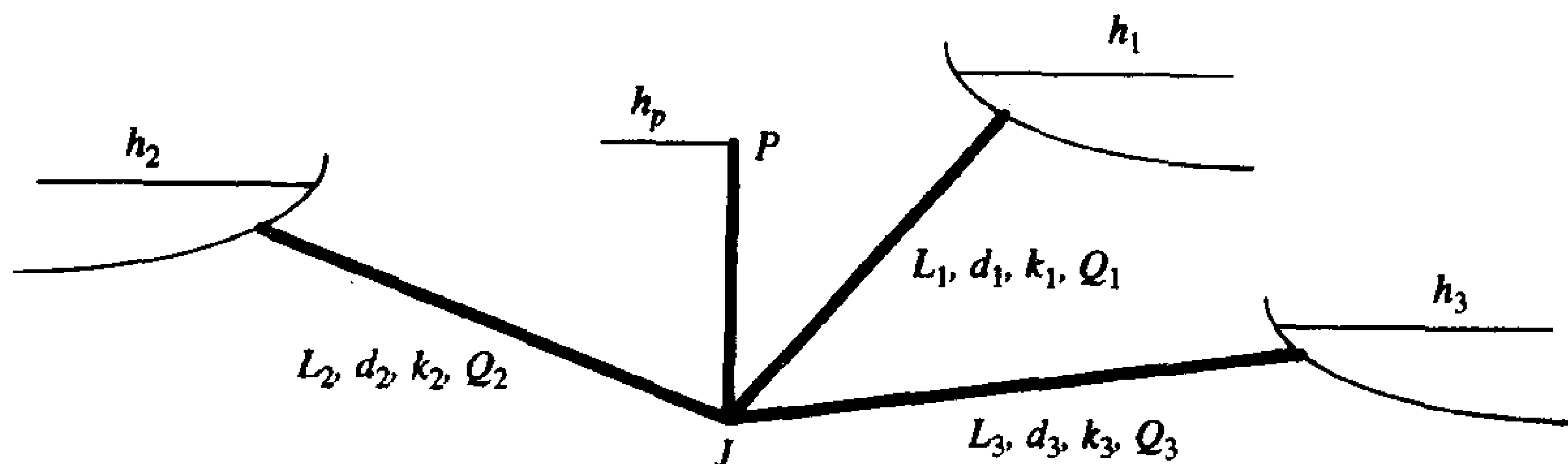


图 11.8　在 J 处连接三个水库的管道

求解方法如下:假定在 h_j 的允许范围内任取一点 h_p,当雷诺数 R_e 非常大时,由式(11.11)求出 λ_j 的假定值,利用这个假定值能计算出每个 V_j 值:

$$\lambda_j = \left[2\lg\left(3.7\frac{d_j}{k_j}\right)\right]^{-2}$$

用 V_j 值求出 R_{ej} 值,然后根据通用的科尔布鲁克公式[参见式(11.11)]再计算 λ_j 的值。重复该过程,直到 V_j 的值处于可接受范围为止。

求出每个 V_j 后,计算 Q_j,并检验是否满足式(11.12)。如果不满足,则选择另一个 h_p 值,计算 V_j 的新值。注意,当 $\Delta h_j = 0$ 时,$Q_j = 0$ 和 $R_{ej} = 0$ 不能计算出 λ_j。在实现该过程的代码和函数中应用了 fzero 函数的嵌套,内部函数 fzero 用来求解 V_j,外部函数 fzero 用来求解 h_p。

下面用一个例子来说明这些结果。

例 11.4　三水库的流动速度

根据以上求解方法,可求出在表 11.2 所示参数下的流速。现创建三个函数,第一个函数为 **ReservoirSumQ**,用于求解 Q_j 和计算式(11.12);第二个函数为 **PipeFrictionCoeff**, 用于计算式(11.11);第三个函数为 **ResFriction**, 用于计算每个 V_j 值下的 λ_j。程序代码如下所示:

```
function Reservoir
d=[0.3,0.5,0.4];
el=[1000,4000,2000];
```

```
k=[0.6,0.6,0.6]* le-3;
h=[120,100,80];
options=optimset('display','off');
hg=fzero(@ReservoirSumQ,110,options,d,el,k,h);
[sq,q]=ReservoirSumQ(hg,d,el,k,h);
disp(['Elevation h_sub_p='num2str(hg)'m'])
disp(['Q1='num2str(q(1))'m^3/s Q2='num2str(q(2))'m^3/s Q3='...
        num2str(q(3))'m^3/s'])

function[sq,q]=ReservoirSumQ(hg,d,el,k,h)
cv=2*9.81*d./el;
ro=d/1.002e-6;
dk=d./k;
qd=0.25*pi*d.^2;
frictguess=(2*log10(3.7*dk)).^-2;
hh=h-hg;
options=optimset('display','off);
for n=1:length(d)
  if hh(n)==0
    q(n)=0;
  else
    lambda=fzero(@ResFriction,frictguess(n),options,dk(n),hh(n),cv(n),…ro(n));
    q(n)=sign(hh(n))*sqrt(cv(n)*abs(hh(n)/lambda)*qd(n));
  end
end
sq=sum(q);

function x=PipeFrictionCoeff(el,re,dk)
if dk>100000|dk==0
  x=el-(2*log10(re*sqrt(el)/2.51))^-2;
else
  x=el-(2*log10(2.51/re/sqrt(el)+0.27/dk))^-2;
end

function lamb=ResFriction(lambda,dk,dh,cv,ro)
ren=sqrt(cv*abs(dh)/lambda)*ro;
lamb=PipeFrictionCoeff(lambda,ren,dk);
```

执行结果为：

```
Elevation h_sub_p=98.904 m
Q1=0.16185 m^3/s  Q2=0.068728 m^3/s  Q3= -0.23058 m^3/s
```

表 11.2 图 11.8 中水库的参数

水库	d_j(m)	L_j(m)	k_j(m)	h_j(m)
1	0.30	1000	0.000 60	120
2	0.50	4000	0.000 60	100
3	0.40	2000	0.000 60	80

11.3 外部流动

11.3.1 无限平面上的边界层由静止突然启动

研究在 x,z 无穷轴平面上 $y=0$ 到 $y=h=10.0$ cm 之间的一层液体。其中，$y=0$ 处是刚

性平板，$y=h$ 处是自由液面。刚开始时，刚性平板和流体处于静止状态。$t=0$ 时，在 x 轴的正方向上平板瞬间加速到速度 U。导致流体的运动仅仅是 x 轴方向，并且是时间和 y 坐标的一个函数。可以求出 U 为 5.0 cm/s 和运动黏度 v_{vis} 为 1.0 cm^2/s 时的函数关系式 $u = u(y,t)$。

对 Navier-Stokes 方程中的 x 部分进行处理可得到这种结果，公式可化简为：

$$\frac{\partial u}{\partial t} = v_{vis}\frac{\partial^2 u}{\partial y^2} \tag{11.13}$$

初始条件是：

$$u(y,0) = 0$$

平板表面的边界条件是滞流条件，为：

$$u(0,t) = U$$

当流体自由面($y=10.0$ cm)的边界条件等于零剪切应力时为：

$$v_{vis}\left.\frac{du}{dy}\right|_{y=10.0} = 0$$

可以用 pdetool 来求解这个问题。在 pdetool 窗口上用 *Options* 菜单中的 *Axes Limits* 创建一个窗口，其尺寸为：x 方向上 -1.5 到 1.5，y 方向上 0.0 到 10.0。然后创建一个矩形，其尺寸为：x 方向上 -0.25 到 0.25，y 方向上 0.0 到 10.0。用 PDE 工具箱在(x,y)域内求解方程，这个计算过程与 x 无关。如果 x 计算量与 y 的计算量相比非常小，则计算时间将会缩短。

矩形的边界条件设定如下：矩形底部(平板的表面)的边界条件为 $u = U = 5.0(h = 1, r = 5)$，矩形顶部(液体的自由面)的边界条件为$\partial u/\partial x = 0(q = g = 0)$。该条件也适用于与 x 无关的矩形的两个垂直边，用改进两次的网格可以求得这个解。为了在 pdetool 中确定式(11.13)，利用 *PDE* 下拉菜单中 *PDE Specification* 窗口选择 *Parabolic* 选项。然后设置 $c = 1$，$a = 0$，$f = 0$ 和 $d = 1$，最后用 *Solve* 下拉菜单的 *Parameters* 选项来确定计算结果的显示时间，在 *Time* 区域上输入 0:0.5:10。解这个方程，在 MATLAB 命令窗口上输出结果 u 和网格参数 p，e 和 t。u 是一个($n_m \times n_t$)的矩阵，其中，n_m 是网格点的个数，n_t 是求解次数，该例中 n_t = 21。以下的程序代码是在 $x = 0$ 上求解的数值，得出如图 11.9 所示的曲线，其中曲线之间的时间间隔 Δt 为 0.5 s：

```
ymin = 0.0; ymax = 10.0;
y = linspace(ymin, ymax, 30);
[nm, nt] = size(u);
for i = 1:nt
  plot(tri2grid(p,t,u(:,i),0,y),y)
  hold on
end
axis([0,5,ymin,ymax])
ylabel('y(cm)')
xlabel('Horizontal velocity (cm/s)')
text(1.5,5,'t = 10.0s')
```

11.3.2 勃拉休斯边界层

在壁面的平面边界层上，如果是不可压缩流场，则其边界层方程为：

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{11.14}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=v_{\text{vis}}\frac{\partial^2 u}{\partial y^2}$$

式中,x 是平行于物体表面的坐标,y 是垂直于物体表面的坐标,u 和 v 分别是相应的流速,v_{vis} 是运动黏度。边界条件为物体平面上的 u 和 v 都为零,且当 $y\to\infty$ 时,$u\to U$①。

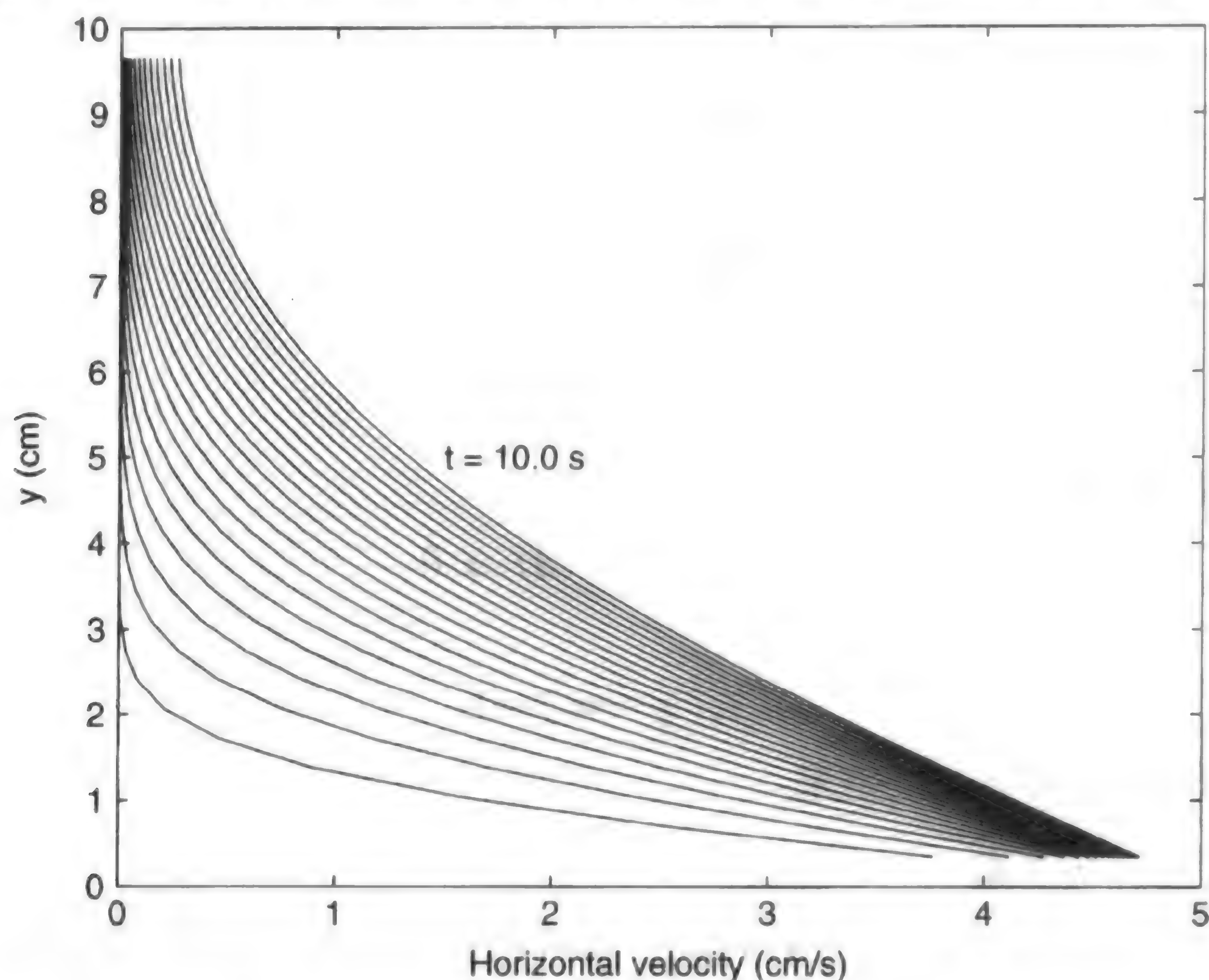


图 11.9　流层 10.0 cm 深处的水平速度突然加速到 $U=5.0$ cm/s

一个类似的解法是:

$$\frac{\mathrm{d}f}{\mathrm{d}\eta}=\frac{u}{U} \tag{11.15a}$$

和

$$\eta=y\sqrt{\frac{U}{xv_{\text{vis}}}} \tag{11.15b}$$

变量 f 与流函数成比例,$\mathrm{d}^2f/\mathrm{d}\eta^2$ 与切应力成比例。该解法可将边界层方程转换为一般非线性微分方程:

$$2\frac{\mathrm{d}^3f}{\mathrm{d}\eta^3}+f\frac{\mathrm{d}^2f}{\mathrm{d}\eta^2}=0 \tag{11.16}$$

式中,当 $\eta=0$ 时,

$$f=0\quad \frac{\mathrm{d}f}{\mathrm{d}\eta}=0 \tag{11.17a}$$

而当 $\eta\to\infty$ 时,

$$\frac{\mathrm{d}f}{\mathrm{d}\eta}\to 1 \tag{11.17b}$$

① 全部细节可参见 R.L.Panton, *Incompressible Flow*, 第二版, John Wiley & Sons, New York, NY, 1996 年, 第 516 页。

为求解式(11.16),引入以下定义将其简化为三个一阶方程:

$$f_1 = f$$
$$f_2 = \frac{\mathrm{d}f}{\mathrm{d}\eta} \tag{11.18}$$
$$f_3 = \frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2}$$

得到:

$$\frac{\mathrm{d}f_1}{\mathrm{d}\eta} = f_2$$
$$\frac{\mathrm{d}f_2}{\mathrm{d}\eta} = f_3 \tag{11.19}$$
$$\frac{\mathrm{d}f_3}{\mathrm{d}\eta} = -0.5 f_1 f_3$$

当 $\eta = 0$ 时,边界条件变为:

$$f_1(0) = f_2(0) = 0 \tag{11.20a}$$

而当 $\eta \to \infty$ 时,

$$f_2(\eta \to \infty) \to 1 \tag{11.20b}$$

下面用一个例子来说明这些结果。

例 11.5 平板的层流边界层

利用 bvp4c 求微分方程式(11.19)的解。求解过程参见 5.5.4 节,程序如下:

```
function BlasiusSolution
solinit = bvpinit(linspace(0,8,8),@BlasiusTguess);
sol = bvp4c(@BlasiusT,@BlasiusTbc,solinit);
x = linspace(0,8);
y = deval(sol,x);
plot(x,y(1,:),'k-',x,y(2,:),'k--',x,y(3,:),'k-.')
xlabel('\eta')
ylabel('f,df/d\eta,d^2f/d\eta^2')
legend('f','df/d\eta','d^2f/d\eta^2','Location','NorthWest')
axis([0,4,0,3])

function dydx = Biasius(x,y)
dydx = [y(2);y(3);-0.5*y(1)*y(3)];

function F = BlasiusT(x,y,flag,Pr)
Pr = 0.07;
F = [Blasius(x,y(1:3));y(5);-Pr*0.5*y(1)*y(5)];

function res = BlasiusTbc(ya,yb)
res = [ya(1);ya(2);ya(4);yb(2)-1;yb(4)-1];

function y = BlasiusTguess(x)
y(1) = x;
y(2) = x^0.5;
y(3) = 5-x;
```

```
y(4) = x/10;
y(5) = 0.5 * (1 - x/10);
```

执行这段程序得到在 $\eta = 0$ 处的切应力为 0.3321，结果如图 11.10 所示。

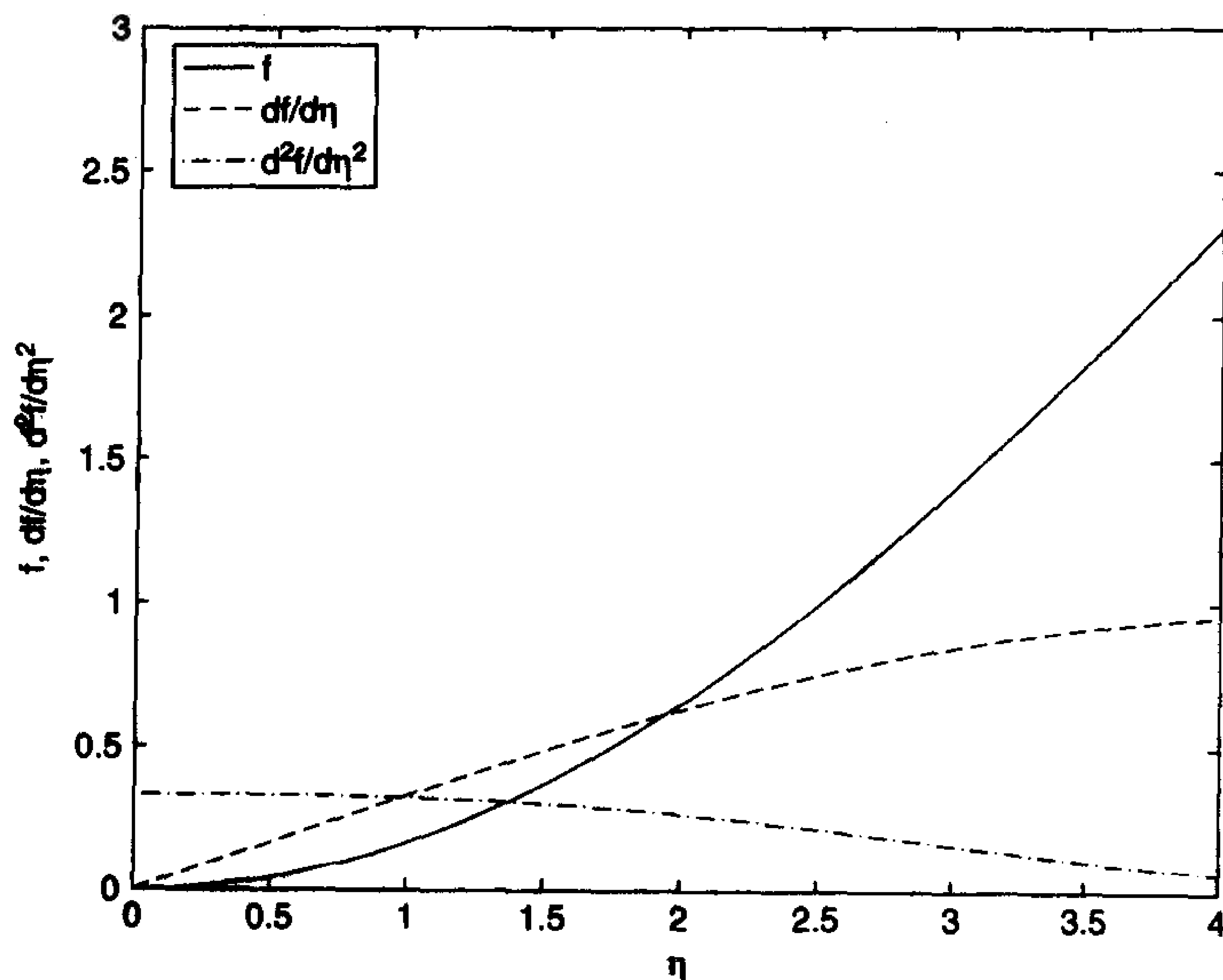


图 11.10　流函数 $f_1 = f$，速度 $f_2 = \mathrm{d}f/\mathrm{d}\eta$ 的流向分量和切应力 $f_3 = \mathrm{d}^2 f/\mathrm{d}\eta^2$ 的勃拉休斯边界层的剖面图

11.3.3　不可压缩势流

对于不可压缩势流，速度场 $\vec{u}$ 受以下条件的限制：

$$\nabla \cdot \vec{u} = 0$$

和

$$\nabla \times \vec{u} = 0$$

这些条件表明速度可表示为势场 ϕ 的梯度：

$$\vec{u} = \nabla\phi$$

式中的 ϕ 满足拉普拉斯方程：

$$\nabla^2\phi = 0 \tag{11.21}$$

利用流函数可以得到二维空间流线簇的另一数学表达式，式中，

$$u = \frac{\partial \Psi}{\partial y}$$

$$v = -\frac{\partial \Psi}{\partial x}$$

流函数也满足拉普拉斯方程。边界条件可以由纽曼条件和第立克特条件组成，与边界正交的速度分量确定纽曼条件，ϕ 的值确定第立克特条件。固体边界处的纽曼条件是 $\vec{u} \cdot \hat{n} = 0$，式中的 $\hat{n}$ 是边界的单位正交。在下文的例子中将讨论几种获得二维空间流线簇的方法，这些方法中有两种是将已知的势函数或流函数叠加在一起来构建流线簇。现给出这样的四个值。

点源和点汇:

$$\phi_M = \frac{m}{2\pi}\ln r_M \qquad \Psi_M = \frac{m}{2\pi}\theta_M$$

$$r_M^2 = (x - x_M)^2 + (y - y_M)^2 \qquad \theta_M = \tan^{-1}\frac{y - y_M}{x - x_M}$$

式中,(x_M, y_M)是点源或点汇的坐标位置,m 是源的强度。

偶极流:

$$\phi_K = \frac{K\cos\theta_K}{r_K} \qquad \Psi_K = -\frac{K\sin\theta_K}{r_K}$$

$$r_K^2 = (x - x_K)^2 + (y - y_K)^2 \qquad \theta_K = \tan^{-1}\frac{y - y_K}{x - x_K}$$

式中,(x_K, y_K)是偶极流的坐标,K 是偶极流的强度。

点涡:

$$\phi_\Gamma = \frac{\Gamma}{2\pi}\theta_\Gamma \qquad \Psi_\Gamma = -\frac{\Gamma\ln r_\Gamma}{2\pi}$$

$$r_\Gamma^2 = (x - x_\Gamma)^2 + (y - y_\Gamma)^2 \qquad \theta_\Gamma = \tan^{-1}\frac{y - y_\Gamma}{x - x_\Gamma}$$

式中,(x_Γ, y_Γ)是旋涡的坐标位置,Γ 是旋涡的强度。

均匀流场:

$$\phi_U = Ux \qquad \Psi_U = Uy$$

式中,U 是流速。

总之,可以将这些不同的流函数结合在一起模拟不同形状的流。如果 Ψ_s 是新的流线函数,则:

$$\Psi_s = \Psi_M + \Psi_K + \Psi_\Gamma + \Psi_U$$

例 11.6　用 contour 绘出流线模型

第一种也是最简单的确定流型方法是用 contour 绘制流线。以下程序画出了一种流体 Ψ_s 的流线图。这种流体由均匀流、偶极流和点涡组成。其中,均匀流速度为 U,偶极流的坐标为(x_K, y_K)且强度为 K,点涡的坐标为(x_Γ, y_Γ)且强度为 Γ。为了绘出该结果,选择偶极流和点涡的坐标为$(-1, -1)$,每个量的强度数值为:

$$K = 5.0 \qquad (x_K, y_K) = (-1, -1)$$

$$\Gamma = 8\pi \qquad (x_\Gamma, y_\Gamma) = (-1, -1)$$

$$U = 5.0$$

如图 11.11 所示,流线是沿着一个圆柱体流动的。绘出该图的主要难点是:如何选择等高线以得到完整描述轮廓的基准面。可以用区域左下角的 Ψ_s 值作为最小值,用区域顶部中间的 Ψ_s 值作为最大值来实现。在图 11.11 中圆柱体的表面(也是一个流线)已经叠加到这些流线上了。

生成图 11.11 的程序代码为:

```
nx = 100;xmin = -3.5;xmax = 1.5;
ny = 100;ymin = -3.0;ymax = 1.5;
```

```
[x y] = meshgrid(linspace(xmin,xmax,nx),linspace(ymin,ymax,ny));
U0 = 5.0;
Gamma = 8 * pi;xGamma = - 1.0;yGamma = - 1.0;
K = 5.0;xK = - 1.0;yK = - 1.0;
radius = inline('sqrt((x - x1).^2 + (y - y1).^2)','x','y','x1','y1');
PsiK = K * sin(atan2(y - yK,x - xK))./radius(x,y,xK,yK);
PsiGamma = Gamma * log(radius(x,y,xGamma,yGamma))/2/pi;
StreamFunction = U0 * y - PsiGamma - PsiK;
levmin = StreamFunction(1,nx);
levmax = StreamFunction(ny,nx/2);
levels = linspace(levmin,levmax,50)';
contour(x,y,StreamFunction,levels)
hold on
theta = linspace(0,2 * pi);
plot(xGamma + cos(theta),yGamma + sin(theta),'k-')
axis equal
axis([xmin,xmax,ymin,ymax])
ylabel('y')
xlabel('x')
```

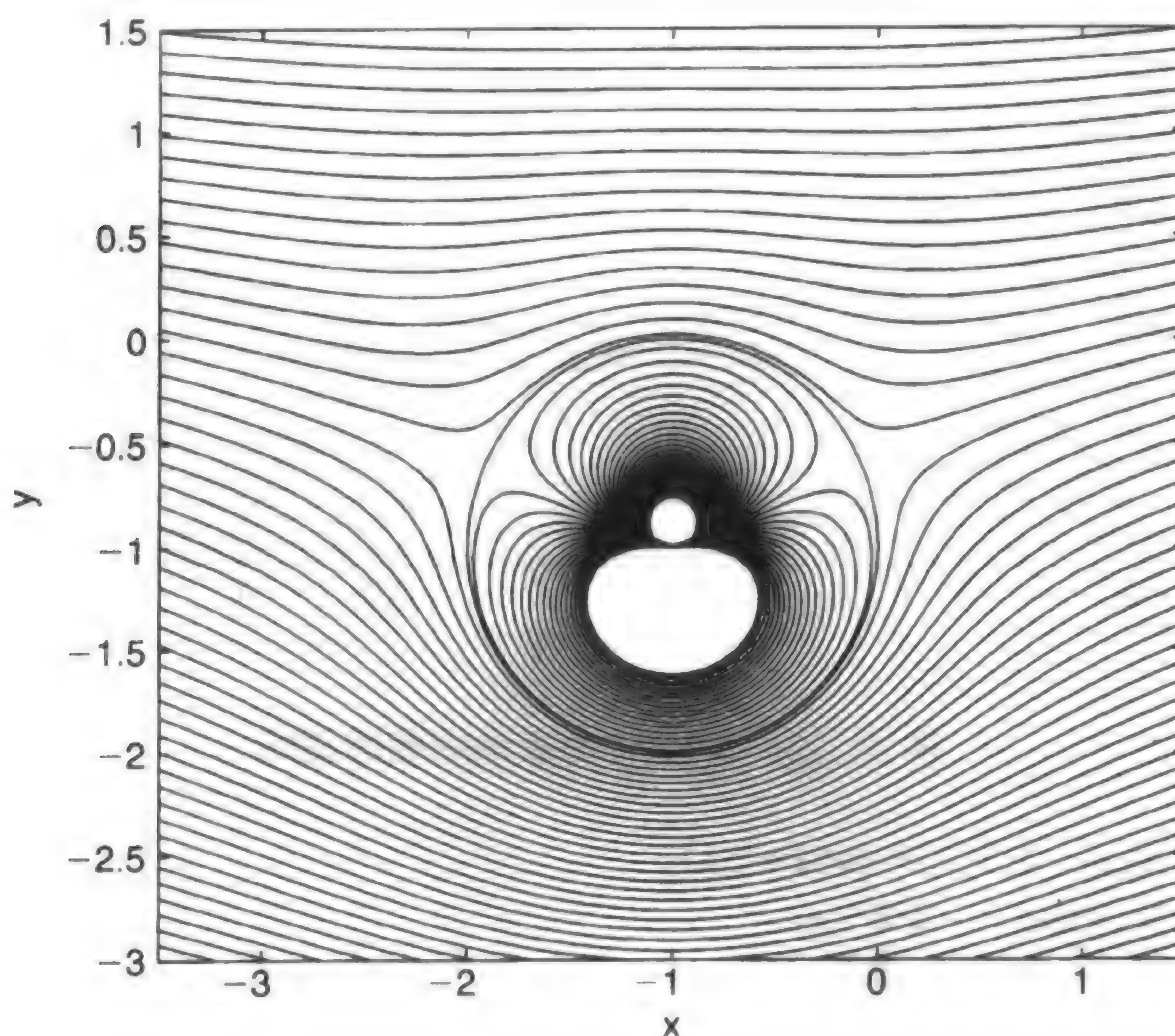


图 11.11 从 Ψ_s 的轮廓线得到交叉流动中环绕圆柱体的流线

例 11.7 直接计算流线

第二种确定流型的方法是用 fzero 确定流线。举一个例子:假定流由均匀流和点源组成,均匀流在 y 轴正方向上 $U = 1$,一个强度 $m = 4.0$ 的点源在(0, -1)处,另一个强度 $m = -4.0$ 的点源在(0, 1) 处。则有:

$$\Psi_{\text{oval}} = +\Psi_U + \Psi_{M_1} + \Psi_{M_2}$$

这些分量可以得出一个椭圆形的均匀流,椭圆方程由下式给出[①]:

① 参见 L.M. Milne-Thomson, *Theoretical Hydrodynamics*, Dover, Mineola, NY, 1996 年, 第 216 页。

$$\frac{2xa}{x^2+y^2-a^2}=\tan\frac{xU}{m/2\pi} \tag{11.22}$$

式中,U 是流速,m 是点源强度,a 是特征尺寸。

在这个例子中使用 fzero 的困难是如何选择一个恰当的初始值。在下面的程序中,通过在直线 $y=-2.0a$ 上的一系列 x 点找到流函数 Ψ_{oval} 的值来求解这个问题。其中给定初始值,在 $y=-2.0a$ 条件下绘制流线,在每个连续的 y 点上,fzero 用 **StreamFun** 函数求出流函数中 x 点的坐标。x 值作为假设值对应前面的 y 值。假定 $a=1$ 时程序输出如图 11.12所示。这幅图旋转 90°就得到了习惯上的水平流。因为是由式(11.12)直接画出来的,所以这里并未得出与椭圆的边界相一致的流线。程序代码为:

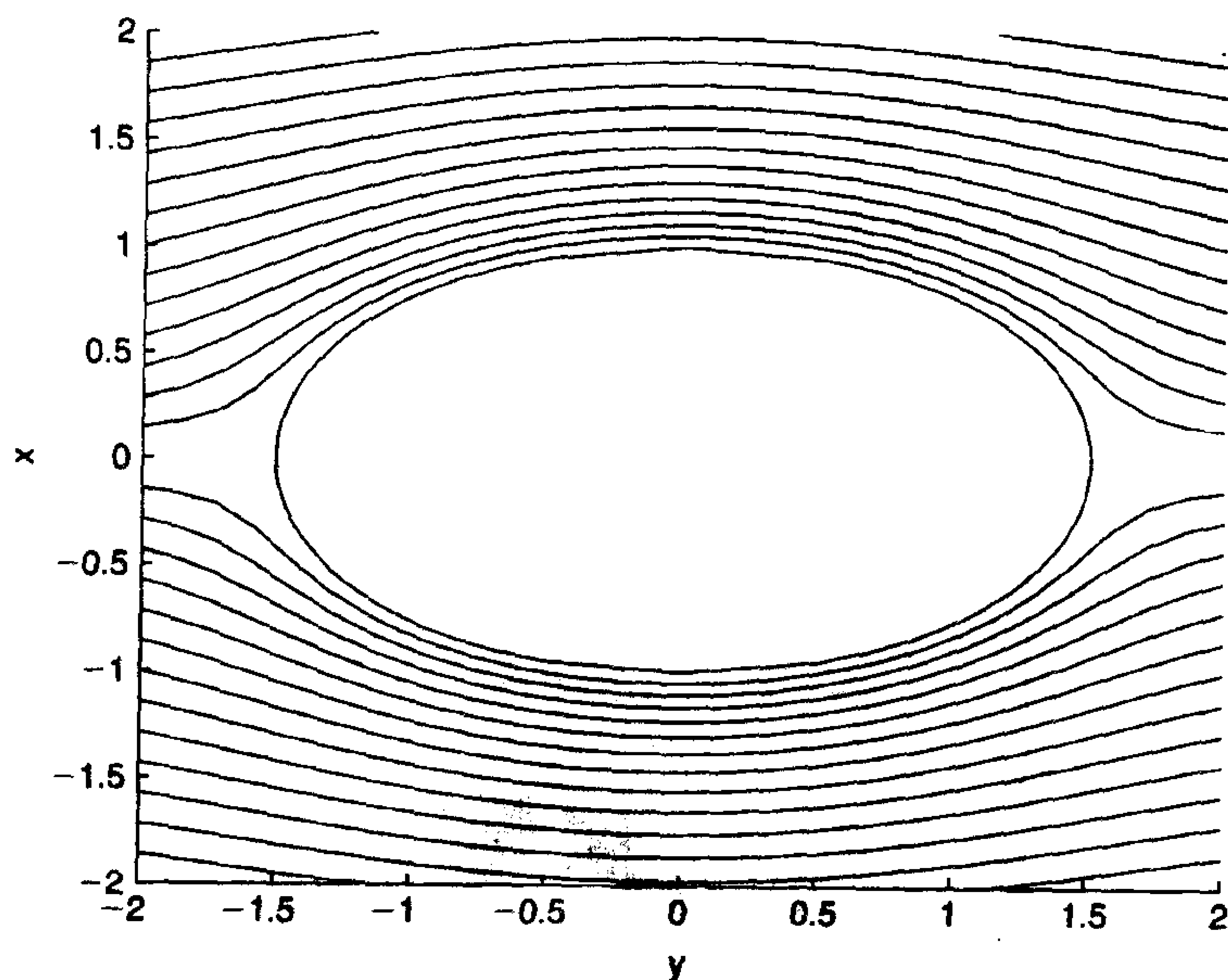


图 11.12 式(11.22)表示的椭圆流线

```
U = 1.0; a = 1.0; m = 4.0; co = m/(2 * pi);
nPsi = 15; n = 30; yStart = -2.0 * a;
xStart = linspace(0,2 * a,nPsi);
y = linspace(-2 * a,2 * a,n);
x = zeros(1,n);
StreamFun = inline ('-U * x - co * (atan2(x,y + a) - atan2(x,y - a)) - psi','x','y',...
                    'psi','U','co','a');
Psi = StreamFun(xStart,yStart,0,U,co,a);
options = optimset('display','off');
for j = 1:nPsi
  guess = xStart(j);
  for i = 1:n
    x(i) = fzero(StreamFun,guess,options,y(i),Psi(j),U,co,a);
    guess = x(i);
  end
  if j > 1
    plot(y,x,'k-',y,-x,'k-')
  end
```

```
    hold on
  end
  axis([ - 2 * a,2 * a, - 2 * a,2 * a])
  ylabel('x')
  xlabel('y')
  xx = linspace( - 1,1,40);
  yy = sqrt(1 - xx.^2 + 2 * xx./tan(xx/co));
  plot(yy,xx,'k', - yy,xx,'k - ')
```

其中,*nPsi* 是流线的条数,*n* 是沿每条流线计算出的点数。

例 11.8 用 pdetool 求解流场

第三种方法是用 pdetool 直接对式(11.21)求解,解决势流问题。一个半径为 1.0 的圆柱体放置在流速为 10.0 的管道中央,现求解该圆柱体的流场问题。在 pdetool 窗口上选定窗口的轴线为:在 x 方向上从 -3.5 到 3.5, 在 y 方向上从 -2.5 到 2.5,然后以(0,0)为中心点、半径为 1.0 做一个圆($C1$),以(0,0)为中心、长度为 6.0、高度为 5.0 做一个矩形。改变 *Set Formula* 使之成为 $R1-C1$,然后选择 *Boundary Mode*。

选定 *Specify Boundary Conditions*,在圆上和矩形的上下边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=0$。为实现该条件,设定 $q=0$ 和 $g=0$。在矩形的左边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=10$。在矩形的右边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=-10$,即设定 $q=0$ 和 $g=-10$。因为在 pdetool 中边界的单位正交 $\hat{n}$ 从边界指向流体区域,所以前面的条件表示流是从左到右的。

接下来初始化网格并对其做两次改进。在 *PDE* 下拉菜单的 *PDE Specification* 上选择 *Elliptic* 来确定偏微分方程。方程式为 $\nabla^2u=0$,该例中 MATLAB 变量 u 代表 ϕ;为此,设定 $c=1, a=0$ 和 $f=0$。求得 u 并在 u 的等高线上绘出速度 ∇u,如图 11.13 所示。

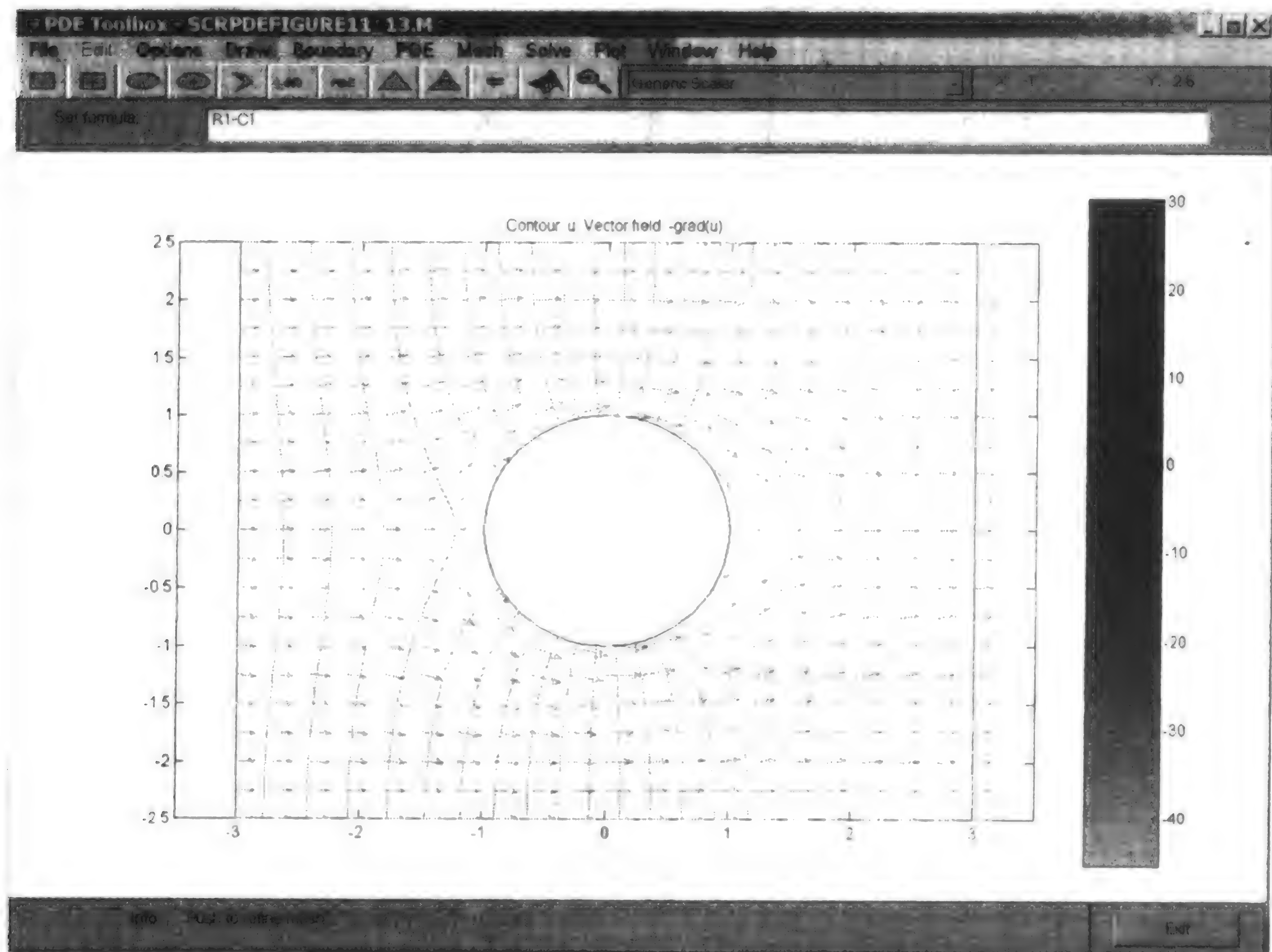


图 11.13 对方程式(11.21)求解得到的速度值(等高线)和方向

为得到更详尽的结果,将 u 的结果和网格坐标 p, e 和 t 的值输出到 MATLAB 命令窗口上。沿着任意一条垂直或水平线的速度曲线用下面的方法都可以得出。分析沿着圆柱体顶部到矩形顶部垂线 $x = 0$ 的水平速度,用以下程序可以绘出速度分布的曲线图。在程序代码中首先创建一个矩形表格,使用 tri2grid 设定 $-0.5 \leqslant x \leqslant 0.5$($n_x = 9$ 点)和 $0.5 \leqslant y \leqslant 2.5$($n_y = 2.5$ 点),然后用 gradient 得到不同区域。最后,通过在 x 方向分隔网格划分适当的差值,从而在直线 $x = 0$ 上得到梯度的水平分量。在程序中速度的结果数组称为 ux,图 11.14给出 ux 与 y 的关系曲线图。对于无限流场中的一个圆柱体,最大速度为 $2U$,出现的位置在圆柱体的顶部和底部(相对于流向 ±90°)。在该例中,将圆柱放在由矩形确定的管道中,圆柱体上最大速度再次出现在 90°处,这个值为 $u_x(1) = 22.64U$。

```
nx = 9;xmin = -0.5;xmax = 0.5;
x = linspace(xmin,xmax,nx);
ny = 25;ymin = 0.5;ymax = 2.5;
y = linspace(ymin,ymax,ny);
uxy = tri2grid(p,t,u,x,y);
[DX,DY] = gradient(uxy);
ux = -DX(:,(nx-1)/2)/((xmax-xmin)/(nx-1));
plot(ux,y,'k-')
axis([10,25,0.9,2.5])
ylabel('y')
xlabel('Horizontal velocity')
hold on
plot([10,25],[1,1],'k-')
text(10.5,1.05,'Top surface of cylinder')
ux1 = max(ux)
```

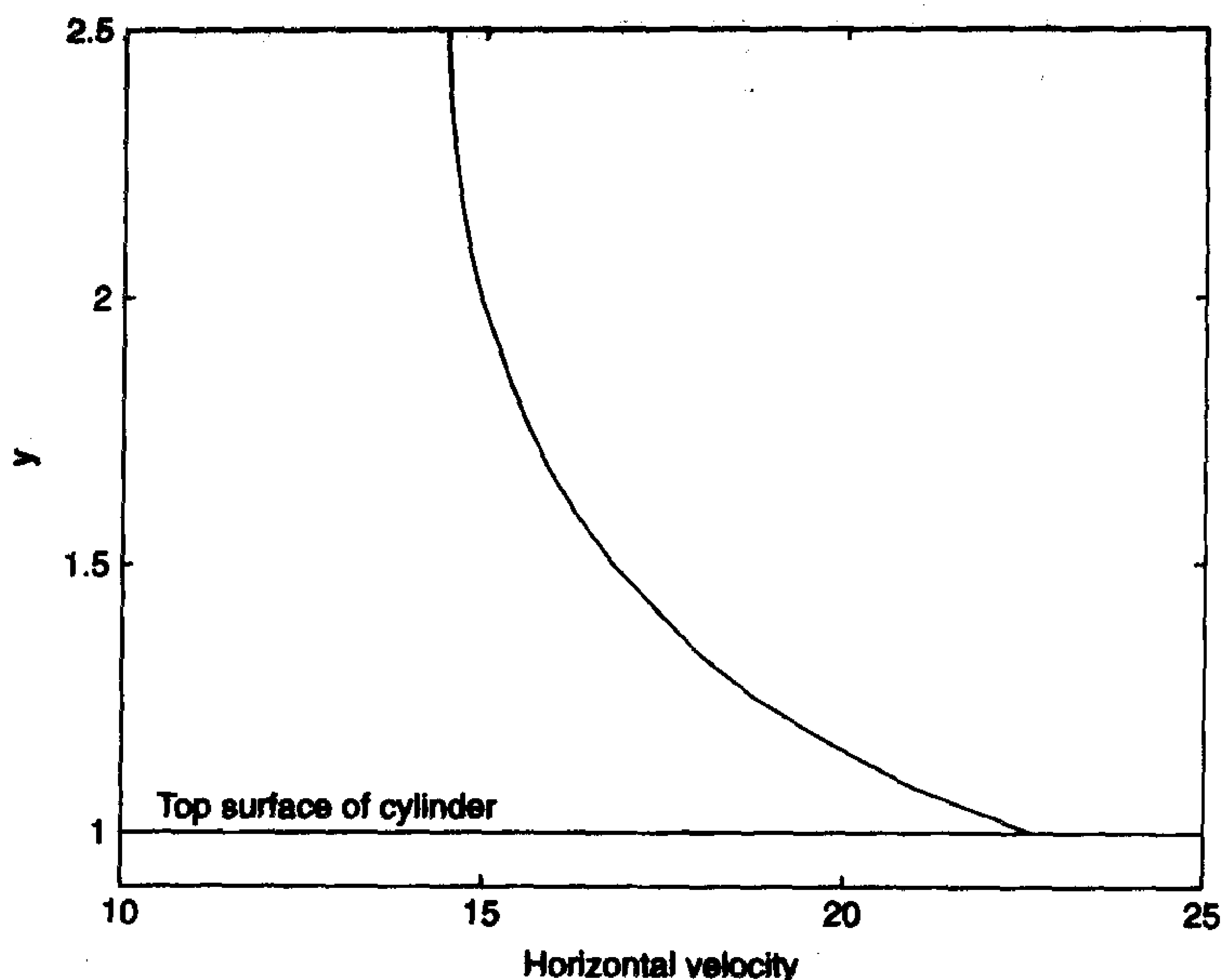


图 11.14 沿圆柱体顶面的垂线方向的水平速度分布

11.3.4 Joukowski 翼型

利用保角映射法可将 z 平面上($z = x + iy$,式中 $i = \sqrt{-1}$)Joukowski 翼型上的势流变换成 ζ 平面($\zeta = \xi + i\eta$)上绕圆柱体运动的流动。均匀来流的速度为 Q,角度为 α,对应于环量为 Γ 的圆柱体,其沿 ξ 轴的截面圆的半径是 R,圆心在 $\zeta_{off} = \xi_{off} + i\eta_{off}$,则绕流的复势是[①]:

$$F(\zeta) = \phi + i\Psi = Qe^{-i\alpha}(\zeta - \zeta_{off}) + Qe^{i\alpha}R^2/(\zeta - \zeta_{off}) + \frac{i\Gamma}{2\pi}\ln[(\zeta - \zeta_{off})/R]$$

式中,ϕ 和 Ψ 分别是速度势和流函数。

如图 11.15 所示,Joukowski 变换由下式给出:

$$z = \zeta + \lambda^2/\zeta \tag{11.23}$$

式中,

$$\lambda = \xi_{off} + \sqrt{R^2 - \eta_{off}^2}$$

是一个实参数,由圆柱体的中心位置确定,相对于 ξ, η 坐标系的原点。Joukowski 变换将 ζ 平面上的每个点映射到 z 平面上的点。依据 $F(\zeta(z))$来变化 F 的值。也就是说,在 ζ 点的 F 值和相应的在 z 点的 F 值是相同的。此变换保持环量和来自远处的圆柱体/翼型的均匀流不变。调整环量,以便使 ζ 平面上圆柱体顺风侧的驻点被移到点$(\lambda, 0)$,此点变成 z 平面上翼型后缘,如图 11.15 中 TE 所标记的。Γ 值可以通过应用 MATLAB 程序来反复绘制圆柱面上流线簇来试验得到,也可以由理论计算得到:

$$\Gamma = 4\pi QR\sin(\alpha - \theta_{TE})$$

程序将在下面给出。(参见图 11.15 中 θ_{TE}的定义,图中 θ_{TE}为负值。)圆柱体或翼型上每单位拱度的升力是:$F_L = \rho Q\Gamma$,式中 ρ 是流体密度。

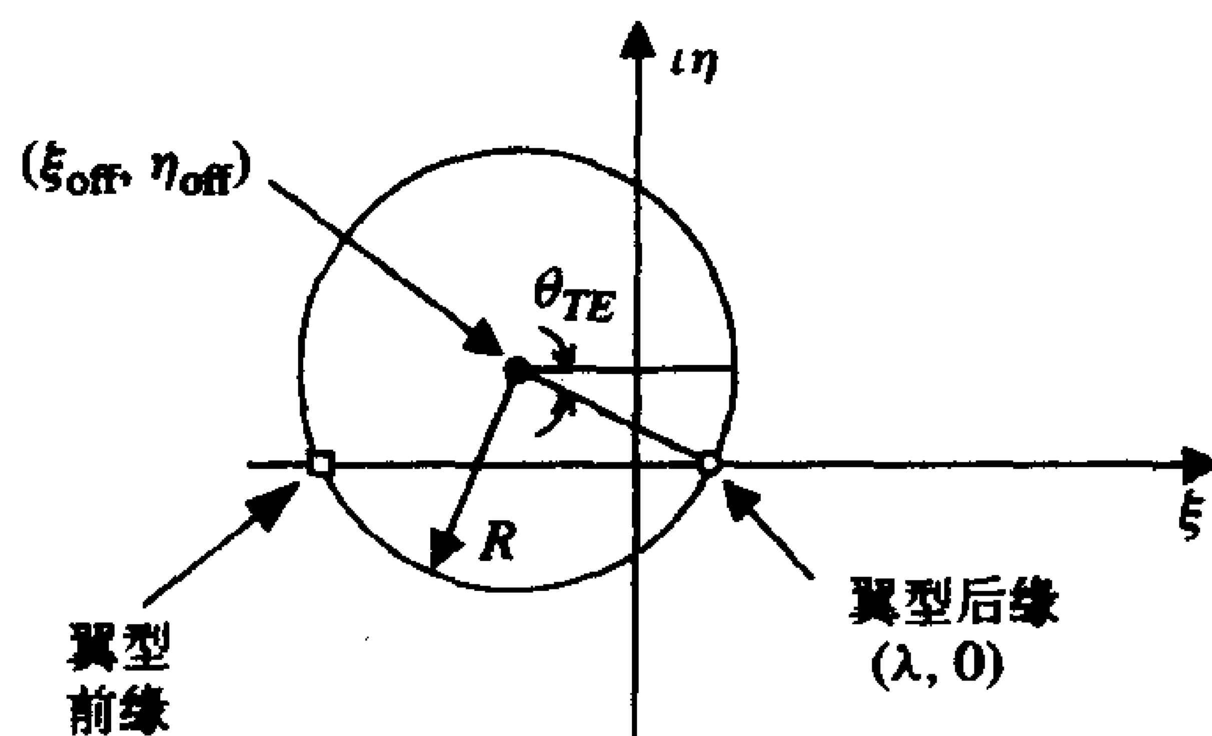


图 11.15 ζ 平面上圆柱体绕流的几何特性

压力是翼型流场中的一个重要量,经常由以下的压力系数代替:

$$C_p = \frac{P - P_\infty}{\rho Q^2/2} = 1 - \frac{q^2}{Q^2} \tag{11.24}$$

式中,P 是局部压力,P_∞是无穷远处的压力,ρ 是流体密度,

$$q = \sqrt{u^2 + v^2}$$

是局部流速,式中 u 和v 分别是流体速度的x 分量和y 分量。z 平面上的局部流速可由以下复

① 参见 R.L.Panton, 出处同上, 第 516 页。

速度计算而得：

$$w = u - iv$$

因此，

$$ww^* = u^2 + v^2 = q^2$$

式中，* 表示复共轭。翼型流场中的复速度可由 F 对 z 的导数获得，如下所示：

$$w = \frac{\mathrm{d}F}{\mathrm{d}z} = \frac{\mathrm{d}F}{\mathrm{d}\zeta}\frac{\mathrm{d}\zeta}{\mathrm{d}z} = \left[Qe^{-i\alpha} - Qe^{i\alpha}R^2/(\zeta - \zeta_{off})^2 + \frac{i\Gamma}{2\pi(\zeta - \zeta_{off})}\right]\frac{\mathrm{d}\zeta}{\mathrm{d}z}$$

式中，

$$\frac{\mathrm{d}\zeta}{\mathrm{d}z} = \frac{1}{1 - \lambda^2/\zeta^2}$$

来自上文中的 Joukowski 变换[参见式(11.23)]。

通过计算流线簇和 Joukowski 翼型周围的压力场的例子来说明上述方程的应用。假定圆柱体的半径 $R = 1.0$ m；而绕流来自与水平方向成 8°角的左下方。圆柱体的偏移是(ξ_{off}, η_{off}) = ($-0.093R$, $0.08R$)。以下程序将首先计算 ζ 平面上的复势 F，并绘制流线簇(Ψ 是常数)，如图 11.16 所示。在图中也标明了与翼型的前缘和后缘相对应的圆柱体上的点。然后将此坐标系平面转换到 z 平面，同时绘制了 z 平面上的流线簇，如图 11.17 所示。最后，计算压力的轮廓线并绘制出来，如图 11.18 所示。所有例子中的轮廓线全用 contour 绘制。由于式(11.23)将圆柱体外部流动变换成翼型的外部流动，而将圆柱体的内部流动变换成整个 z 平面，这样使程序变得复杂了。因此，如果仅获得要求的流线簇和翼型外部压力轮廓线，则可使用以下方法：对于流线簇，将圆柱体内部流函数的值设置为其表面流函数的常数值。换句话说，圆柱体表面是一条流线，且其流函数是常数。对于压力系数来说，由于压力系数的值是沿着圆柱体表面变化的，所以对该方法必须加以修正。因此，内部压力系数的值设置成 0。

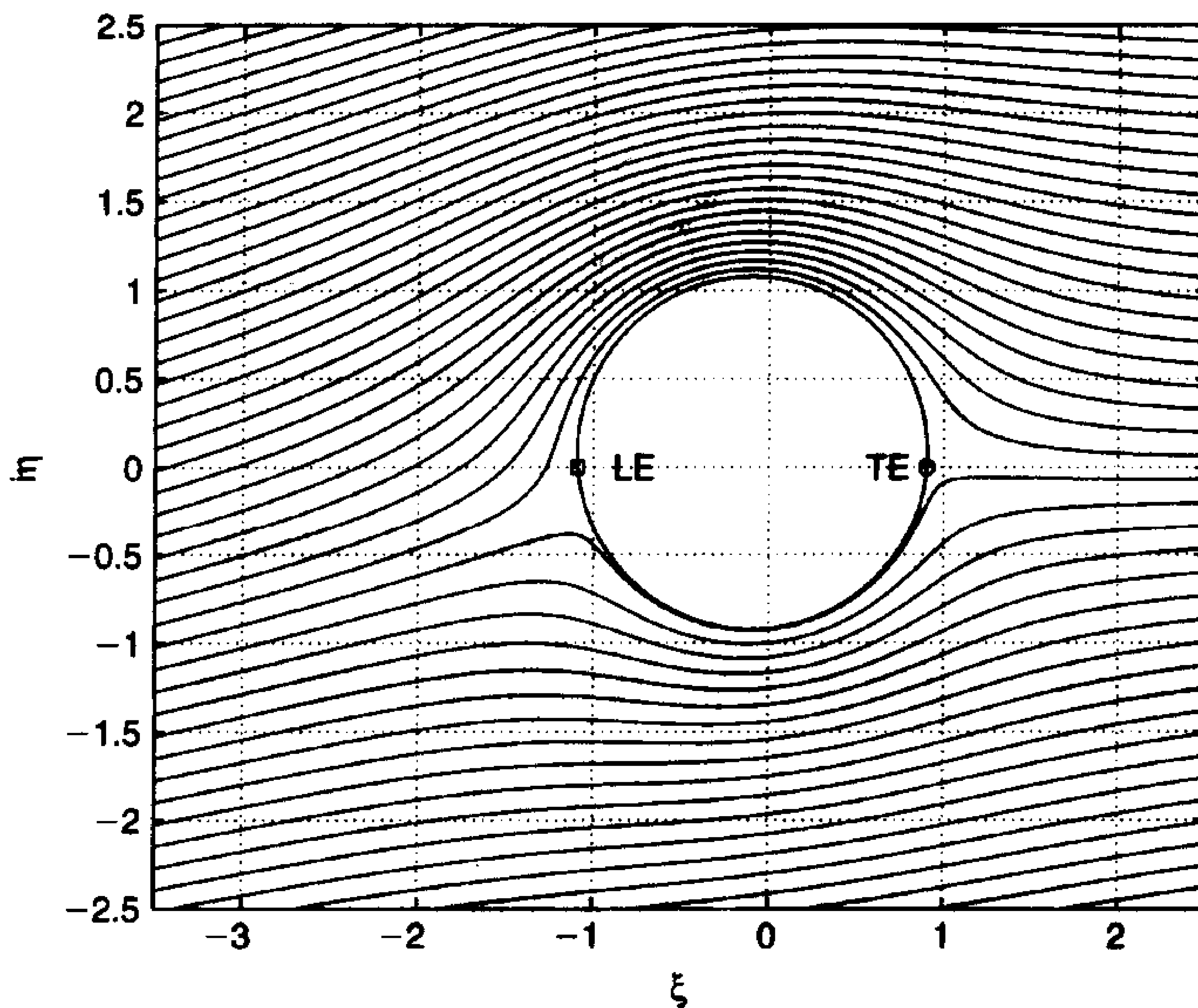


图 11.16 ζ 平面上绕环量是 Γ 的圆柱体的流线模型，圆柱半径 $R = 1\text{m}$, $\alpha = 8°$, (ξ_{off}, η_{off}) = ($-0.093R$, $0.08R$)

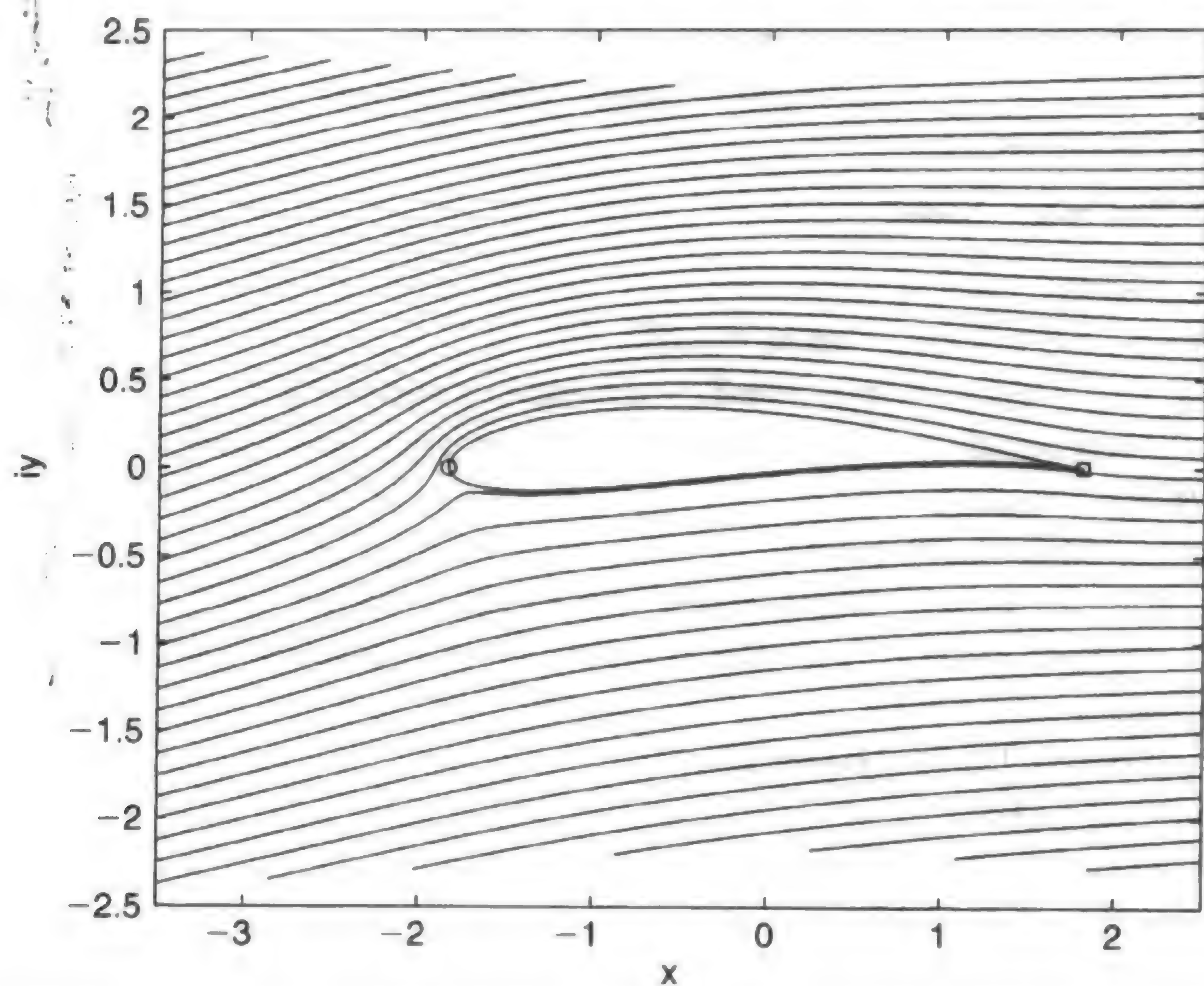


图 11.17 通过变换图 11.16 中所示的圆柱和流型得到 Joukowski 翼型和流线簇。图中的矩形和圆分别对应图11.16中的矩形和圆。此时，$R=1\mathrm{m}$，$\alpha=8°$，$(\xi_{\mathrm{off}},\eta_{\mathrm{off}})=(-0.093R,0.08R)$

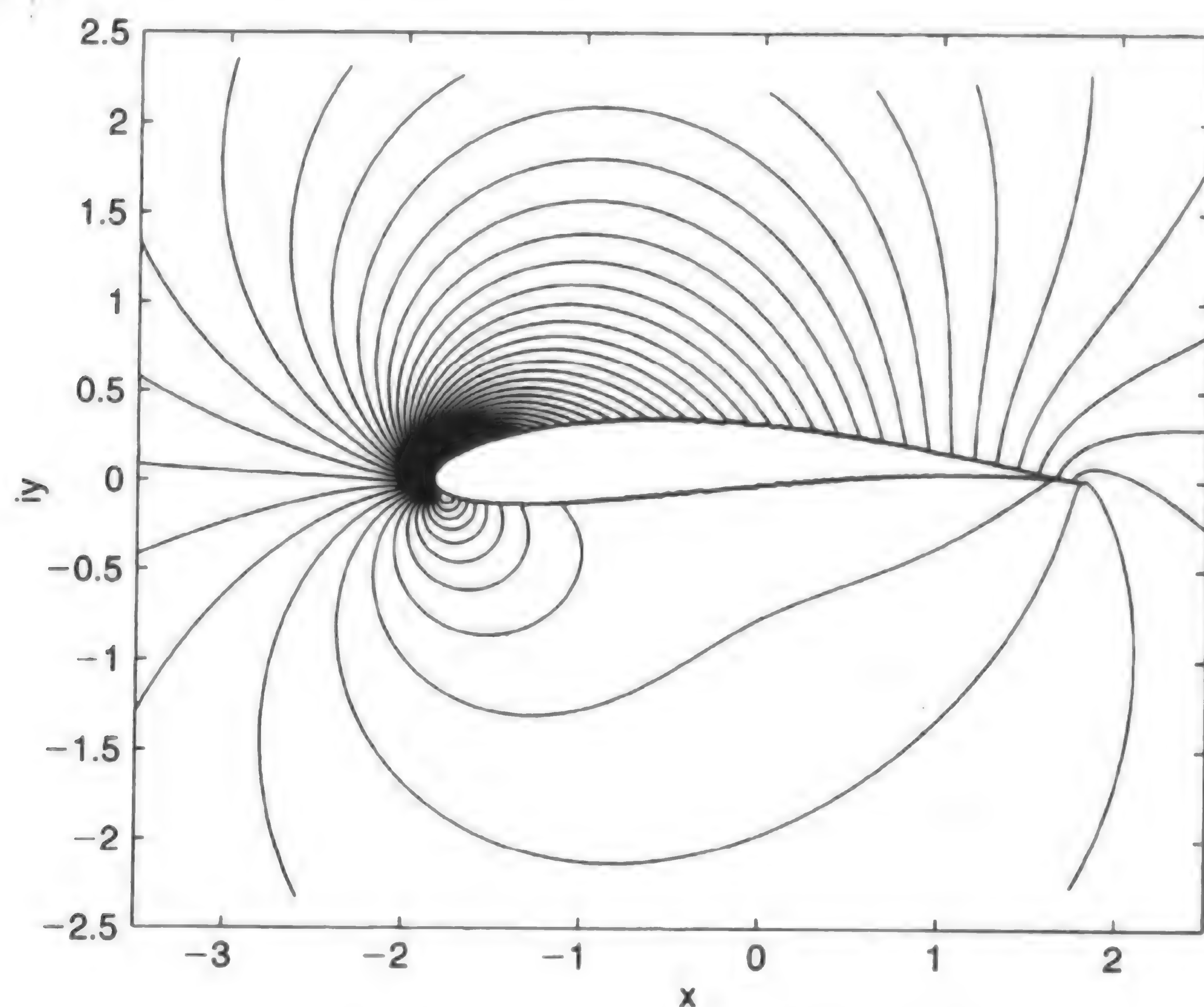


图 11.18 Joukowski 翼型附近的固定压力系数的轮廓图。此时，$R=1\mathrm{m}$，$\alpha=8°$，$(\xi_{\mathrm{off}},\eta_{\mathrm{off}})=(-0.093R,0.08R)$

图 11.16、图 11.7 及图 11.18 的程序如下：

```
R = 1.0; Q = 1.0; alpha = 8.0 * pi/180;
ksioff = -0.093 * R; etaoff = 0.08 * R; zetaoff = complex(ksioff, etaoff);
nksi = 800; ksimin = -3.5 * R; ksimax = 2.5 * R;
neta = 800; etamin = -2.5 * R; etamax = 2.5 * R;
```

```
[ksi,eta] = meshgrid(linspace(ksimin,ksimax,nksi),...
                     linspace(etamin,etamax,neta));
zeta = complex(ksi,eta);
figure(1) %Flow over cylinder
thetaTE = - asin(etaoff/R);
Gamma = 4 * pi * Q * R * sin(alpha-thetaTE);
F = Q * exp ( - i * alpha) * (zeta-zetaoff) + Q * exp(i * alpha) * R^2./
        (zeta-zetaoff) + ...i * Gamma/(2 * pi). * log((zeta-zetaoff)/R);
StreamFunction = imag(F);
zetapt = complex(R + ksioff, etaoff);
Fpt = Q * exp( - i * alpha) * (zetapt-zetaoff) + Q * exp(i * alpha) * R^2./
        (zetapt-zetaoff) + ...i * Gamma/(2 * pi). * log(zetapt-zetaoff)/R);
StreamFunctionpt = imag(Fpt);
rad = sqrt((ksi-ksioff).^2 + (eta-etaoff).^2);
indx = find(rad <= R);
StreamFunction(indx) = StreamFunctionpt;
levmin = StreamFunction(1,nksi);
levmax = StreamFunction(neta,1);
levels = linspace(levmin,levmax,50);
contour(ksi,eta,StreamFunction,levels)
axis equal
grid
axis([ksimin,ksimax,etamin,etamax])
xlabel(' \ xi')
ylabel('i \ eta')
hold on
theta = linspace(0,2 * pi,1000);
zetac = R * exp(i * theta) + zetaoff;
plot(zetac,'k-')
hold on
ksiTE = ksioff + sqrt(R^2 - etaoff ^ 2);
ksiLE = ksioff - sqrt(R^2 - etaoff ^ 2);
plot(ksiTE,0,'or')
plot(ksiLE,0,'sr')
text(ksiTE - 0.35 * R,0,'TE')
text(ksiLE + 0.2 * R,0,'LE')
figure(2) %Joukowski airfoil
lambda = ksioff + sqrt(R^2-etaoff ^ 2);
zeta = complex(ksi,eta);
z = zeta + lambda^2./zeta;
x = real(z);
y = imag(z);
contour(x,y,StreamFunction,levels)
axis equal
axis([ksimin,ksimax,etamin,etamax])
xlabel('x')
ylabel('iy')
hold on
zair = zetac + lambda^2./zetac;
xair = real(zair);
yair = imag(zair);
```

```
[xle,ile] = min(xair);
[xte,ite] = max(xair);
plot(zair,'k')
zetaTE = complex(ksiTE,0);
zetalE = complex(ksiLE,0);
zTE = zetaTE + lambda^2/zetaTE;
zLE = zetaLE + lambda^2/zetaLE;
plot(zTE,0,'or')
polt(zLE,0,'sr')
chord = real(zTE-zLE);
figure(3) %Pressure field around airfoil
w = (Q * exp( - i * alpha) - Q * exp(i * alpha) * R^2./(zeta-zetaoff).^2 + ...
             i * Gamma/(2 * pi)./(zeta-zetaoff))./(1.0-lambda^2./zeta.^2);
Cp = 1.0 - w. * conj(w)/Q^2;
Cp(indx) = 0.0;
levels = linspace( - 10,1,150);
contour(x,y,Cp,levels)
hold on
zair = zetac + lambda^2./zetac;
xair = real(zair);
yair = imag(zair);
[xle,ile] = min(xair);
[xte,ite] = max(xair);
plot(zair,'k-')
axis equal
axis ([ksimin,ksimax,etamin,etamax])
xlabel('x')
ylabel('iy')
```

11.4 明槽流

研究如图11.19所示的具有不变棱柱形断面的明槽渐变流和流向坡面 S_0。从槽底往上任意流向位置 z 处的水深用 y 来表示。槽内水的截面积为：

$$A = by(2 + y/(bm))$$

和湿周为：

$$P = 2(b + y\sqrt{1 + 1/m^2})$$

因此，水力半径为：

$$R_h = A/P = \frac{by(2 + y/(bm))}{2(b + y\sqrt{1 + 1/m^2})}$$

槽内体积流量用 Q 表示。两个重要的水深是均匀流水深 y_0 和临界流水深 y_c①。均匀流水深，就是水深和流动条件不随槽的长度变化的深度。这个水深可以通过解曼宁(Manning)方程式得到，此方程式如下：

① 参见 J.B.Franzini 和 E.J.Finnemore, *Fluid Mechanics with Engineering Applications*, McGraw-Hill, New York, NY, 1997年，第427~449页。

$$Q = 1.486 \frac{AR_h^{2/3}S_0}{n} \tag{11.25}$$

式中,S_0 是槽的流向坡面,n 是常数。上式对英制单位是有效的;对于公制单位,应将式中的 1.486 换成 1.0。临界流水深就是弗劳德(Froude)数($v/\sqrt{gA/B}$,式中 v 是平均流速,B 是表面宽度($2b + 2y_c/m$))等于 1 时相应的水深。依据流量,其条件为:

$$Q^2 = g\frac{A^3}{B} = g\frac{(by_c(2 + y_c/(bm))^3}{2b + 2y_c/m} \tag{11.26}$$

式中,g 是重力加速度。对于流量恒定且沿着具有均匀坡面 S_0 槽的非均匀流条件,水深 y 由以下的一阶微分方程确定[①]:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = S_0\frac{1 - (y_0/y)^{3.333}}{1 - (y_c/y)^3} \tag{11.27}$$

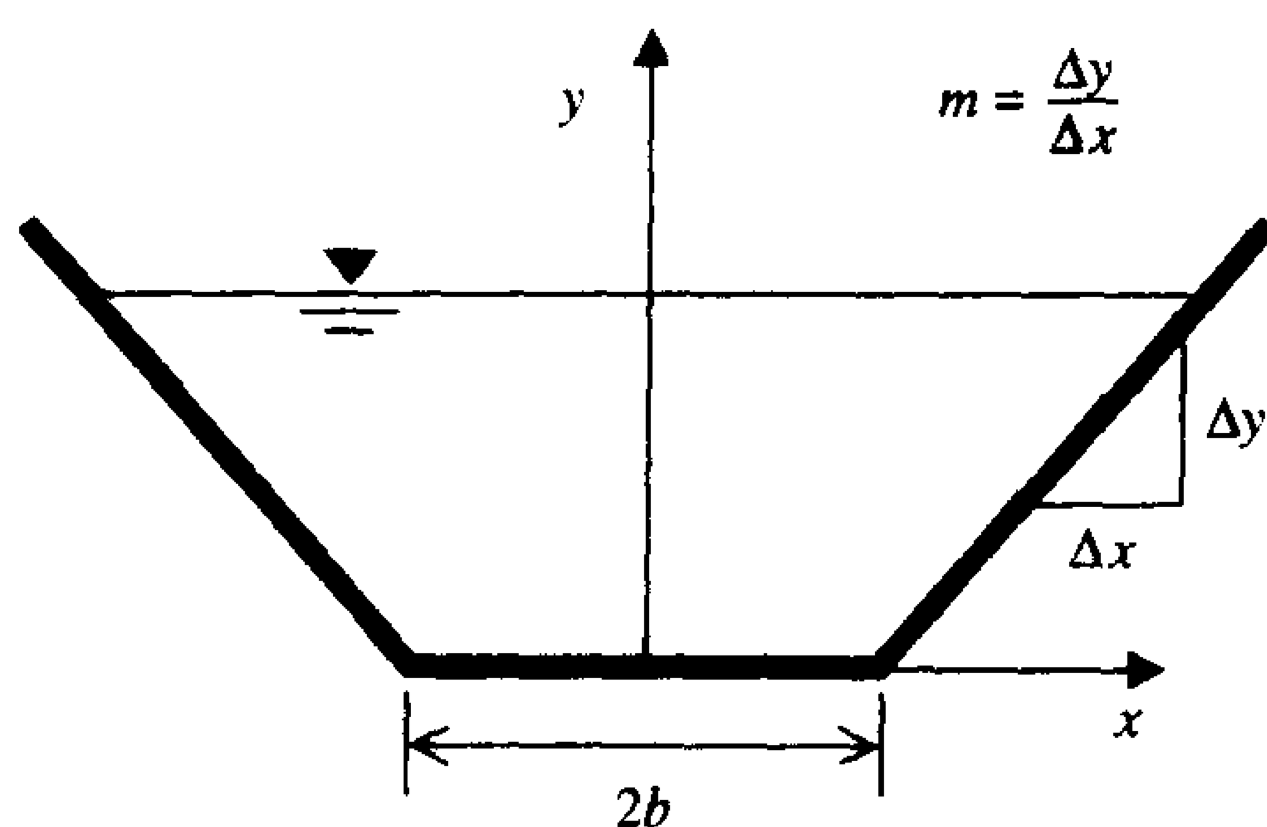


图 11.19 水深为 y 的棱柱形槽的断面

下面用两个例子来说明这些结果的应用。

例 11.9 溢水的均匀槽

研究一个均匀槽,其参量为 $Q = 1000$ ft/s,$n = 0.025$,$m = 0.6667$,$b = 10.0$,四个不同的坡面分别为 $S_0 = 0.0010, 0.0015, 0.0020$ 和 0.0025。本例中,槽的出口是个自由的溢出口。当流动达到临界条件,即 $y = y_c$ 时,在槽的顺流末端出现溢水。槽内水面高度将随着溢水逆流的距离而上升,并最终达到均匀流水深 y_0。对于每个槽坡面,沿着槽的溢水逆流的水面高度的轮廓面将达到点 $y = 0.975y_0$。通过对式(11.27)求积分可解决这个问题。然而,当溢水 $y \to y_c$ 时,右边趋于无穷,这对于转化方程式和对 $\mathrm{d}x/\mathrm{d}y$ 从 $0.975y_0$ 积分到 y_c 最为便利。程序如下,结果如图 11.20 所示。

```
function OpenChannel
grav = 32.2; m = 0.6667; b = 10.0;
n = 0.025; Q = 1000; nS = 4;
slopearr = linspace(0.0010,0.0025,nS);
xmin = -3000; ylegend = 3.0; dyl = 0.5;
opt = optimset('display','off');
for iS = 1:nS
  slope = slopearr(iS);
  y0 = fzero(@Manning,6,opt,Q,n,b,m,slope);
```

① 参见 F.M.Henderson, *Open Channel Flow*, MacMillan Publishing Company, New York, NY, 1966 年,第 131 页。

```
    A_0=b*y0*(2+y0/(b*m));
    P_0=2*(b+y0*sqrt(1+1/m^2));
    Rhd_0=A_0/P_0;
    yc=fzero(@Q26,3,opt,Q,grav,b,m);
    [y,x]=ode45(@dchannel,[yc,0.975*y0],0,[],yc,y0,slope);
    plot(x,y,'k-')
    axis([xmin,0,0,8])
    xlabel('x(m)')
    ylabel('Water depth,y(m)')
    hold on
    text(-1550,ylegend,['S_0='num2str(slope,'%6.4f')...',
            y_0='num2str(y0,4)])
    nx=size(x);
    plot([-1500,x(nx(1)-3)],[ylegend,y(nx(1)-3)-0.1],'k-1')
    ylegend=ylegend-dyl;
end

function dydx=dchannel(y,x,yc,y0,slope)
dydx(1)=1/slope*(1-(yc/y(1))^3)/(1-(y0/y(1)))^3.333);

function A=Manning(y,Q,n,b,m,slope)
A=Q-1.486/n*b*y*(2+y/(b*m))*(b*y*(2+y/(b*m))...
            /(2*(b+y*sqrt(1+1/m^2))))^0.667*slope^0.5;

function B=Q26(y,Q,grav,b,m)
B=Q^2-grav*(b*y*(2+y/(b*m)))^3/(2*b+2*y/m);
```

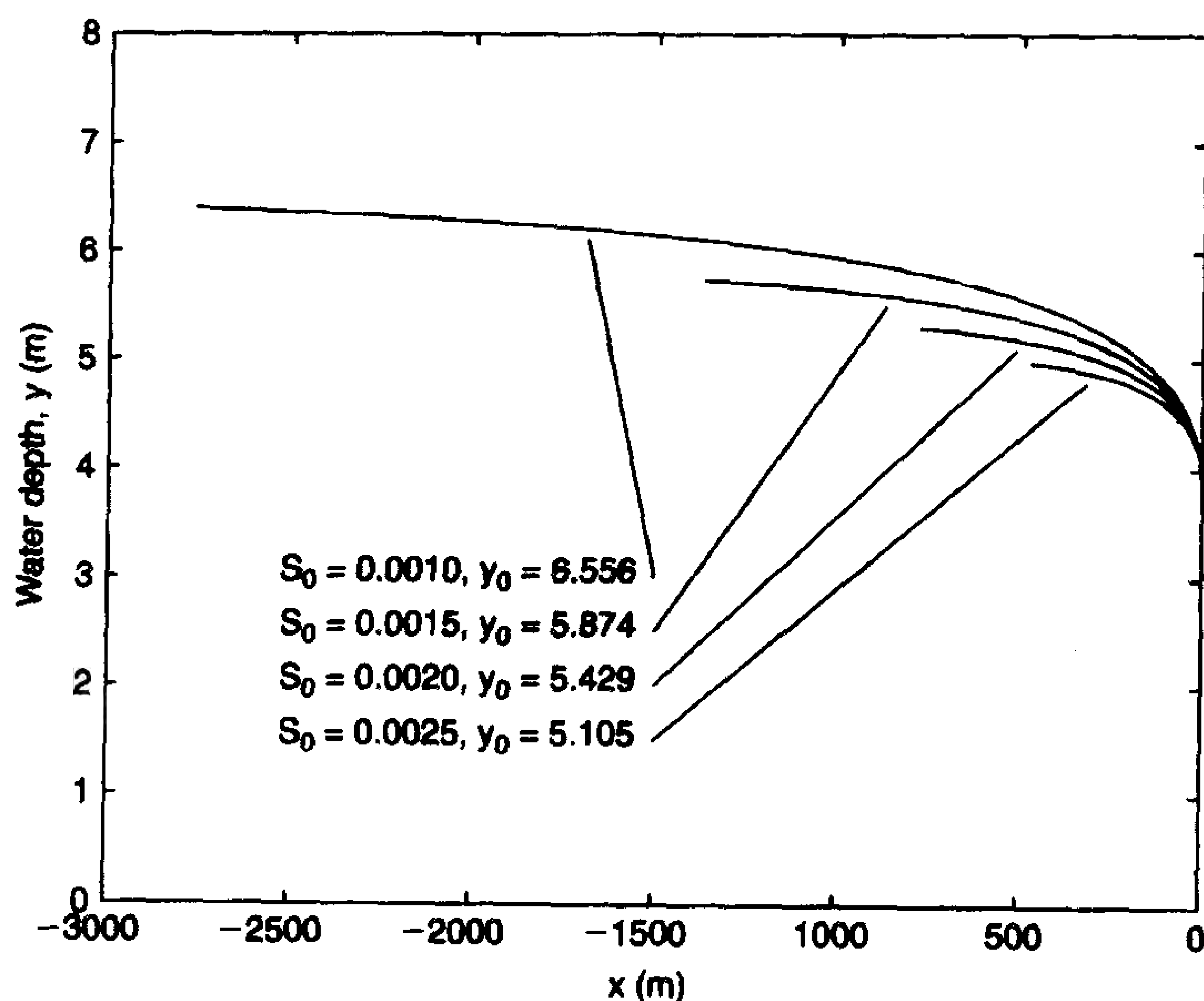


图 11.20 棱柱形槽内水高和距离的关系，其中参量为 $m=0.6667$，$b=10.0$ ft，$n=0.025$，$Q=1000\ \mathrm{ft^3/s}$，并有几个 S_0 值。在 $x=0$ 处溢水自由流出，此处流动到达临界条件

例 11.10 水库流量

如图 11.21 所示，水库通过堰连到一个恒定坡面的沿长度均匀的棱柱形槽上。水库中心处的水面到堰的高度变化是 y_R。应用断面比能方程式计算流量，计算公式为：

$$E = y + v^2/(2g)$$

依据伯努利方程,当沿着自由液面流线从水库中心到堰嘴上方自由液面内的点时,断面比能是恒定的,水库中心处的流速是 0,堰嘴处高出堰的水深是 y_w。在水库中心有 $E = E_0 = y_R$。因此,断面比能方程式简化为:

$$y_R = y_w + v_w^2/(2g)$$

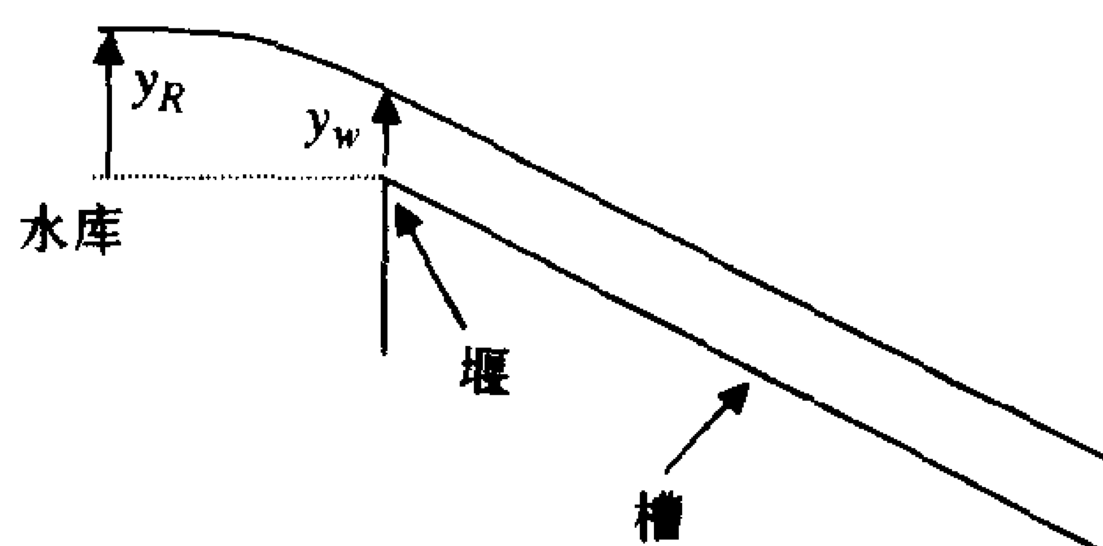


图 11.21　水库和流量槽的示意图

依据流量 Q,有:

$$y_R = y_w + Q^2/(2gA^2) = y_w + Q^2/(2gb^2y_w^2(2 + y_w/(bm))^2) \tag{11.28}$$

当弗劳德数 $v/\sqrt{gA/B}$ 等于 1 时出现堰上的临界水深 y_{wc}。因此,在临界条件下,断面比能方程式变成:

$$y_{wc} + \frac{A_{wc}}{2B_{wc}} = y_R$$

式中,

$$A_{wc} = by_{wc}(2 + y_{wc}/(bm))$$

$$B_{wc} = 2b + 2y_{wc}/m$$

满足曼宁方程[式(11.25)]和断面比能方程[式(11.28)]的 Q 值是所有 $y_w > y_{wc}$ 的流量。当槽的坡面上升时,Q 也随之上升,一直到达堰,也就是到临界条件。对于较高的槽坡面,流量在临界值保持常数。

已知 $y_R = 10.0$ ft, $n = 0.014$, $m = 30.0$, $b = 5.0$ ft 和 $S_0 = 0.001$ 计算 Q 和 y_w。以下关于曼宁方程和断面比能方程的程序执行结果及曲线如图 11.22 所示。本例中,堰上流动条件是亚临界的,两条曲线的交点给出了流量和水深。

```
function Discharge
yr = 10.0;g = 32.2;b = 5.0;m = 30;S0 = 0.001;n = 0.014;
yw = linspace(0,yr,100);
Qbernoulli = sqrt(2 * g * b^2 * (yr-yw). * (2 * yw + yw.^2/(b * m)).^2);
plot(Qbernoulli,yw,'k - ')
hold on
xlabel('Volume flow rate(ft^3/s)')
ylabel('y _ w(ft)')
tex(500,2,'Bernoulli Equation')
A = b * yw. * (2 + yw/(b * m));
P = 2 * (b + yw * sqrt(1 + 1/m^2));
Rh = A./P;
Qmanning = 1.486 * A. * (Rh.^0.6666 * sqrt(S0))/n;
plot(Qmanning,yw,'k - ');
hold on
```

```
text(180,3,'Manning Equation')
opt = optimset('display','off');
ywflow = fzero(@Q2526,[8,5,9,5],opt,b,m,S0,n,g,yr)
Awflow = b * ywflow * (2 + ywflow/(b * m));
Qflow = Awflow * sqrt(2 * g * (yr-ywflow))
text(260,8.9,'Solution(Subcritical Flow)')
text(260,8.2,['Q = 'num2str(Qflow,4)'ft^3/s'])
text(260,7.3,['y_{w} = 'num2str(ywfolw,4)'ft'])
plot(Qflow,ywflow,'ks')
ywcrit = fzero(@Q26,6,opt,b,m,yr);
Awcrit = b * ywcrit * (2 + ywcrit/(b * m));
Qcrit = Awcrit * sdqrt(2 * g * (yr-ywcrit));
hold on
text(690,7,'Critical Flow Point')
text(690,6.3,['Q_{c} = 'num2str(Qcrit,4)'ft^3/s'])
text(690,5.5,['y_{wc} = 'num2str(yncrit,4)'ft'])
plot([Qcrit,ywcrit,'ks'])

function A = Q2526(y,b,m,S0,n,g,yr)
A = 1.486 * (b * y * (2 + y/(b * m))) * (((b * y * (2 + y/(b * m)))...
    /(2 * (b + y * sqrt(1 + 1/m^2))))^0.6666 * sqrt(S0))/n-...
    sqrt (2 * g * b^2 * (yr-y) * (2 * y + y^2/(b * m))^2);

function B = Q26(ywc,b,m,yr)
B = ywc + 0.5 * b * ywc * (2 + ywc/(b * m))/(2 * b + 2 * ywc/m) - yr;
```

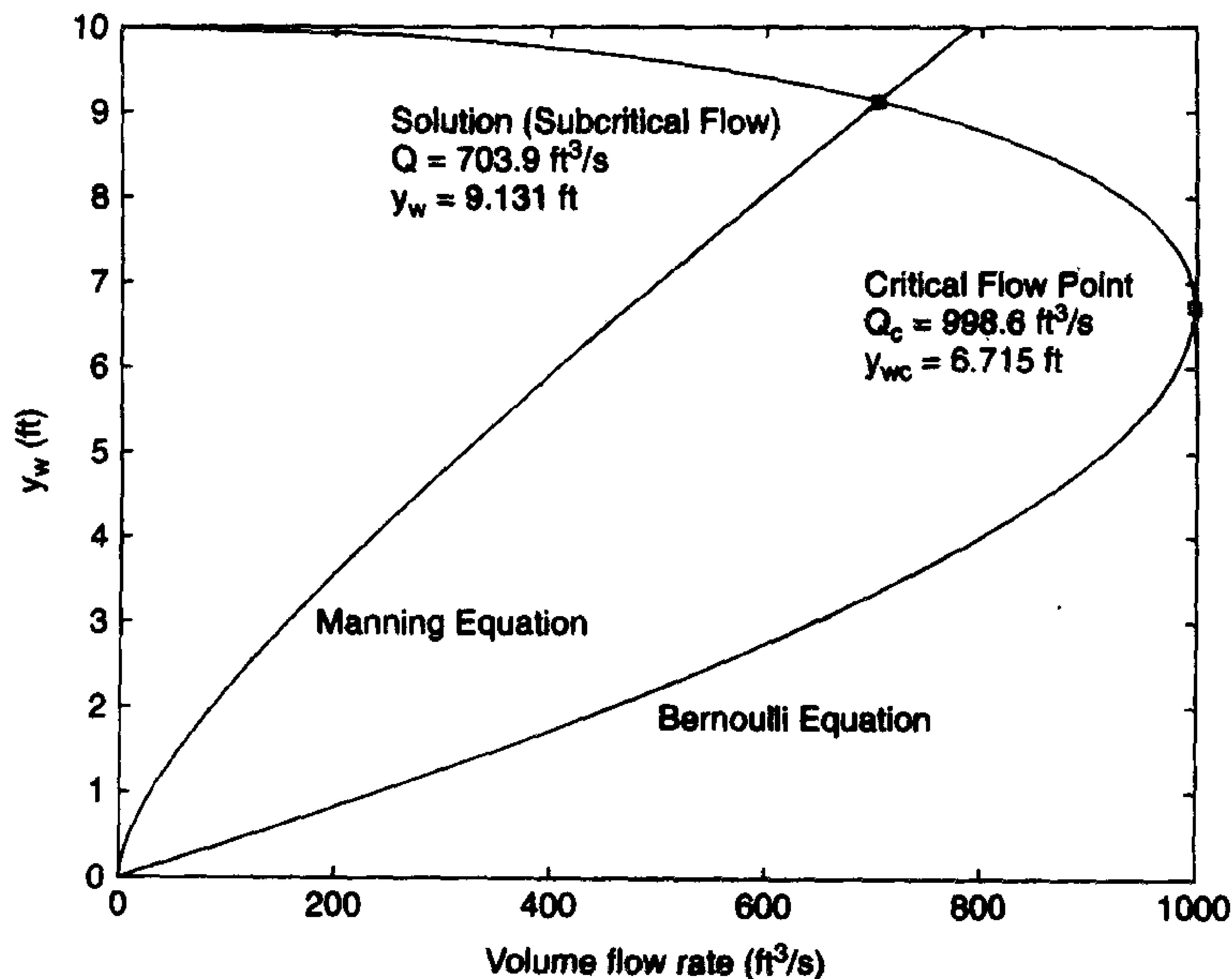


图 11.22 已知 $y_R = 10.0$ ft, $b = 5.0$ ft, $m = 30.0$, $S_0 = 0.001$ 和 $n = 0.014$ 时水库流量的计算

11.5 应用函数一览

本章中所用到的函数参见表 11.3。

表 11.3　第 11 章中所用的 MATLAB 函数

MATLAB 函数	说明
gradient	数值梯度
pdeintrp	在 pdetool 中,从结点数据到三角形中点数据插值
pdetrg	在 pdetool 中,用网格形成每个三角形的面积
trigrid	在 pdetool 中,从三角网格到矩形网格的插值

练习

11.1 计算具有相同横截面积的两条管道中区域的流场,其中一条管道的截面为正方形,另一条管道的截面为矩形。矩形截面的一条边长是另一条边的四倍。用 pdetool 函数比较两管道体积流量。控制方程由式(11.7)给出。设式(11.7)的右边对于每条管道都是相同的,定为 1.0,即有相同的流体和压强梯度。管壁的边界条件为 $u=0$,在 MATLAB 命令窗口中输出每个结果,用 11.2.1 节中的求解过程可以得出:

$$Q_{sq}/Q_{rect} \approx 2$$

式中,Q 为流速。

11.2 在一条高 $2h$ 的水道中通过压强作用的平稳层流,如图 11.23 所示。速度由下式给定:

$$u(y) = -\frac{h^2}{2\mu}\frac{\mathrm{d}P}{\mathrm{d}x}\left(1-\frac{y^2}{h^2}\right)$$

式中,$\mathrm{d}P/\mathrm{d}x$ 是压强梯度,μ 是流的动力黏度。用 pdetool 对该速度分布和另一个管道的速度分布进行比较。其中,管道的高度与水道相同,宽度是水道高度的两倍。假设有相同的压强梯度和黏性系数。求解的微分方程由式(11.7)给定。其结果应与图 11.24 所示相同。

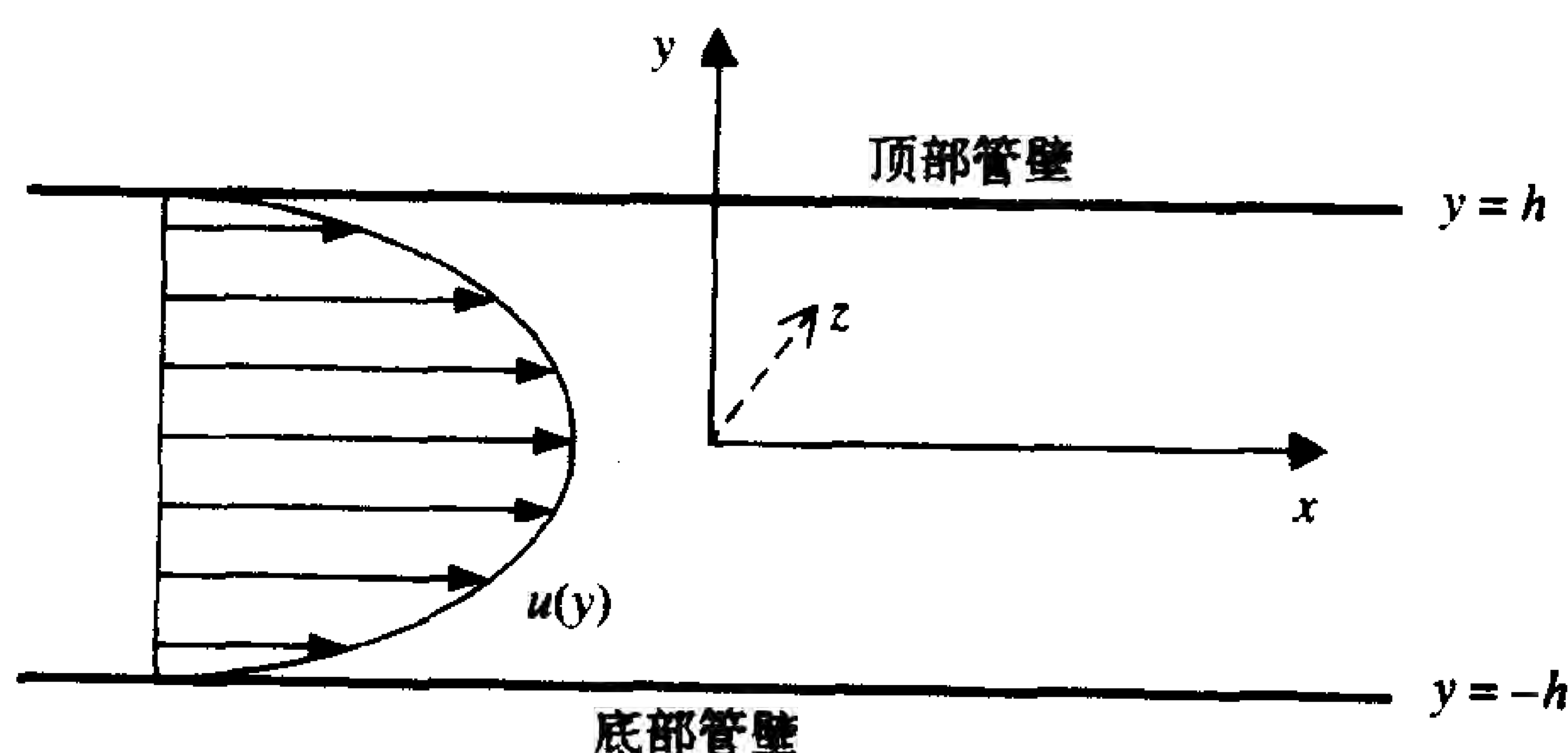


图 11.23　水道沿 z 轴无限延伸(对于管道,在 $z = \pm 2h$ 处放置两个垂直壁)

11.3 研究在一条管道中层流压强驱动充分发展且平稳的油的流动情况,如图 11.25 所示。该管道的横截面为矩形,高为 1 cm,宽为 0.5 cm。油在压强梯度(在图 11.25 中,方向指向书页外面)作用下流动,在流动方向上右壁以 0.5 m/s 的速度沿流动方向移动,而其他三个平面壁静止不动。假定,压强梯度为 10 kPa/m,油的动力黏度为 0.1 kg/m.s。用 pdetool 绘出管道中心处水平和垂直平面的速度分布情况。求解的微分方程由式(11.7)给定。其结果应与图 11.26 和图 11.27 一致。

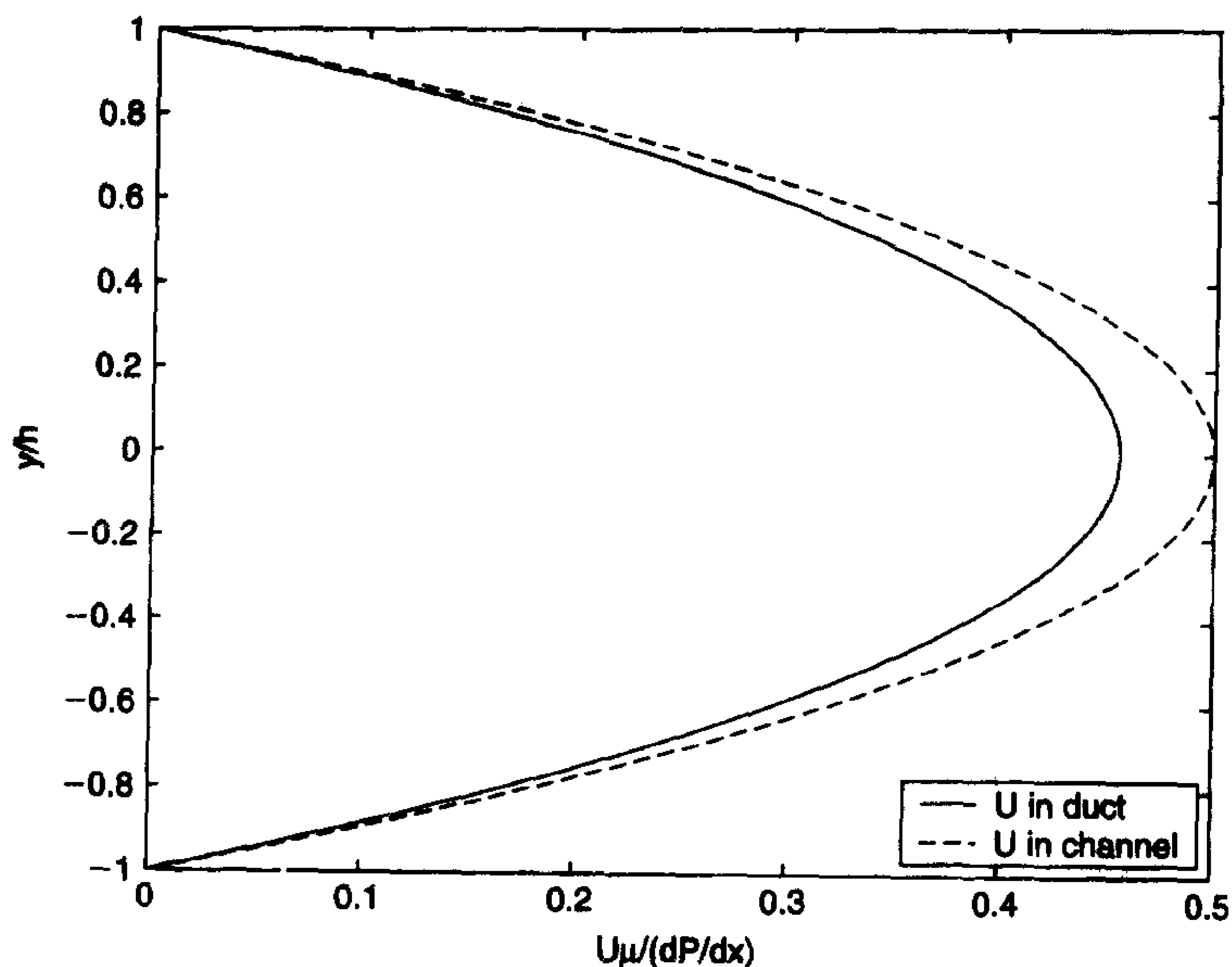


图 11.24 沿管道中心线的速度分布与同样高度水道中速度分布的比较

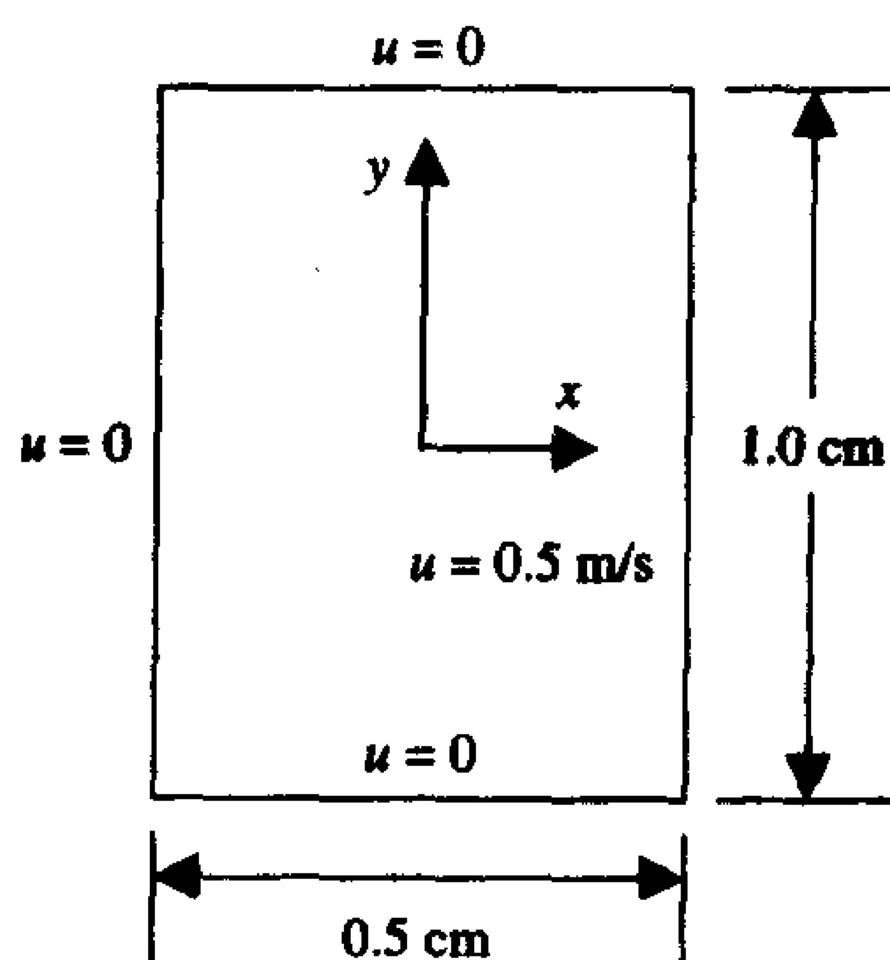


图 11.25 习题 11.3 中管道的横截面

11.4 分析在长和宽分别为 0.3 cm × 0.2 cm 的矩形管道中一种黏性流体，其 μ = 0.02 kg/m.s，ρ = 800 kg/m^3。开始时流体静止，当 t = 0 时，管道轴向压强梯度突然达到 10 kPa/m。在 0 到 0.04 s 时间段内(步长为 0.005 s)，绘出连接管道平面壁 0.3 cm长的中心平面的速度等高线图形。其结果应与图 11.28 所示一致。

11.5 研究通过一条截面为矩形的无缝管道的黏性流体，管道的矩形截面长和宽都是 a。当时间 t = 0 时，流体以 1 m/s 均匀速度通过截面。当 $t > 0$ 时，应用壁的滞流边界条件和轴向压强梯度 dP/dx①。这样，经过长时间后的稳态流速 Q_t 就等于 t = 0 时刻的流速率 Q_0：

$$\frac{dP}{dx} = \frac{28.46\mu Q_0}{d^4}$$

① 参见 F.M. White，*Fluid Mechanics*，第四版，McGraw-Hill，New York，NY，1999 年，第 365 页。

式中，μ 是流体的动力黏度。假定，$a = 1$ cm，流体是 $\mu = 0.2$ kg/m.s，$\rho = 800$ kg/m^{3}的油。利用从 pdetool 得到的结果绘出各个时刻通过管道中心面轴向流速的分布情况，以及管道中流速与时间的关系图。其结果应分别对应于图 11.29 和图 11.30。

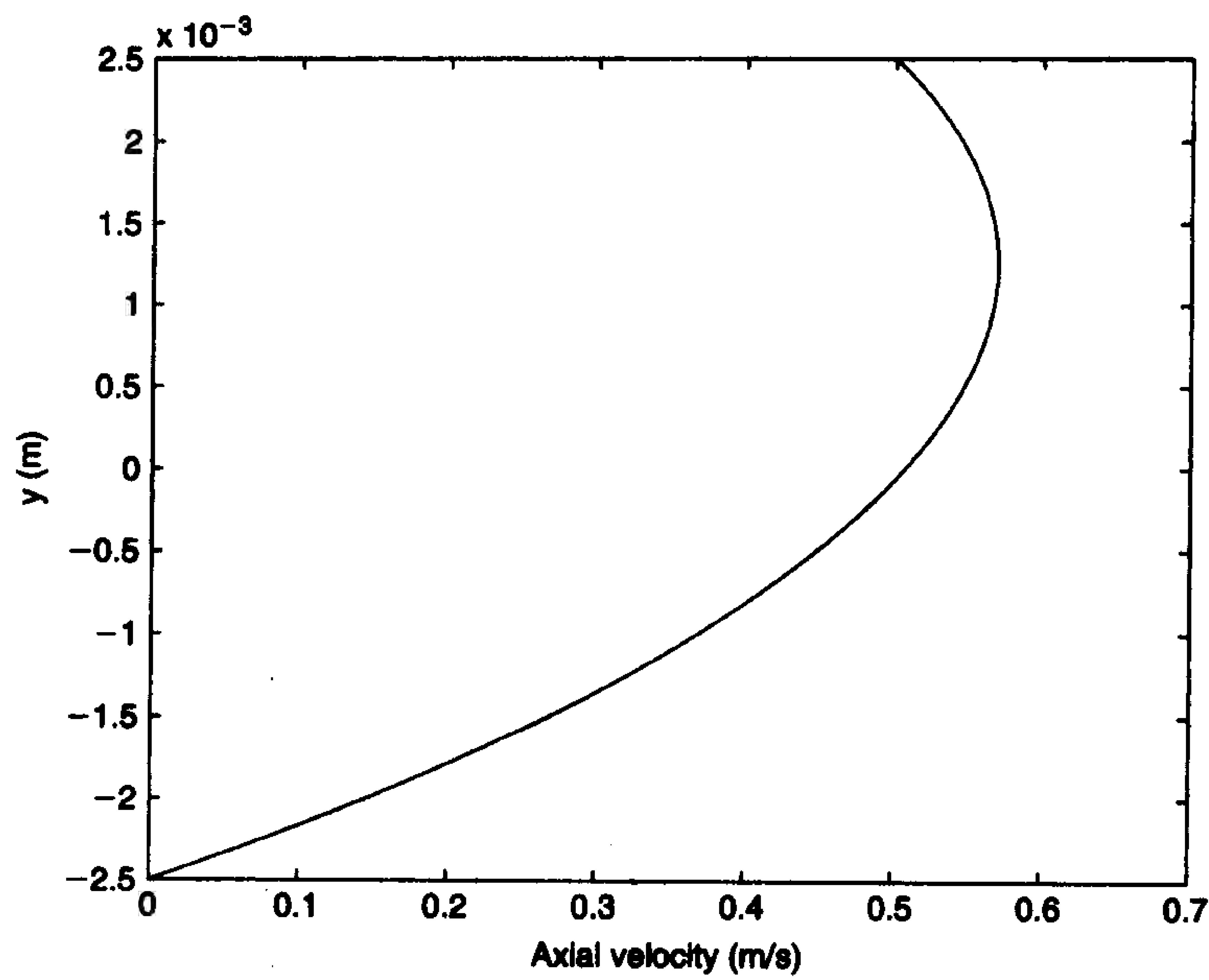


图 11.26 连接固定壁和 1 cm 长移动壁的中心面轴向速度

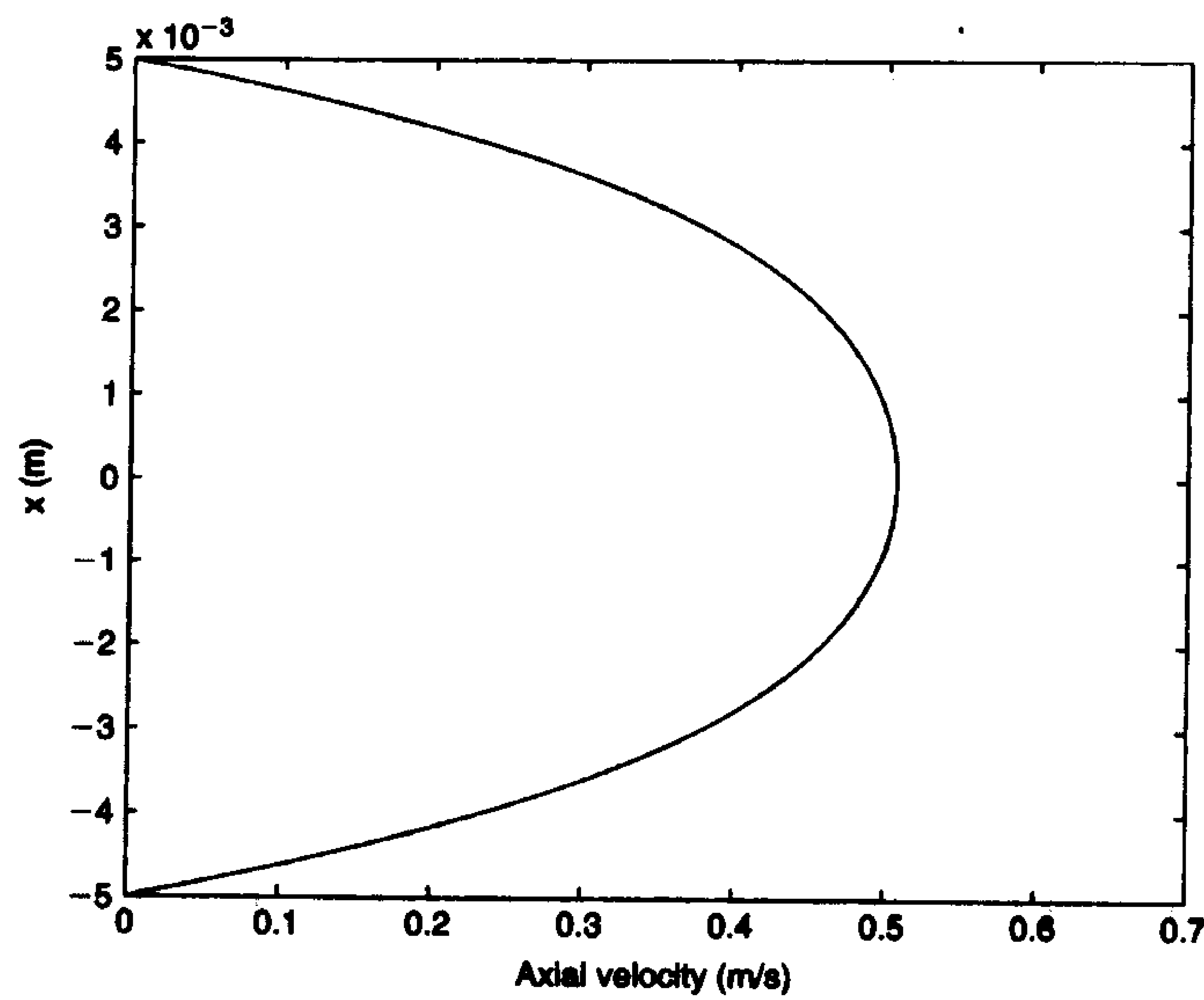


图 11.27 连接两个 0.5 cm 长固定壁的中心面轴向速度

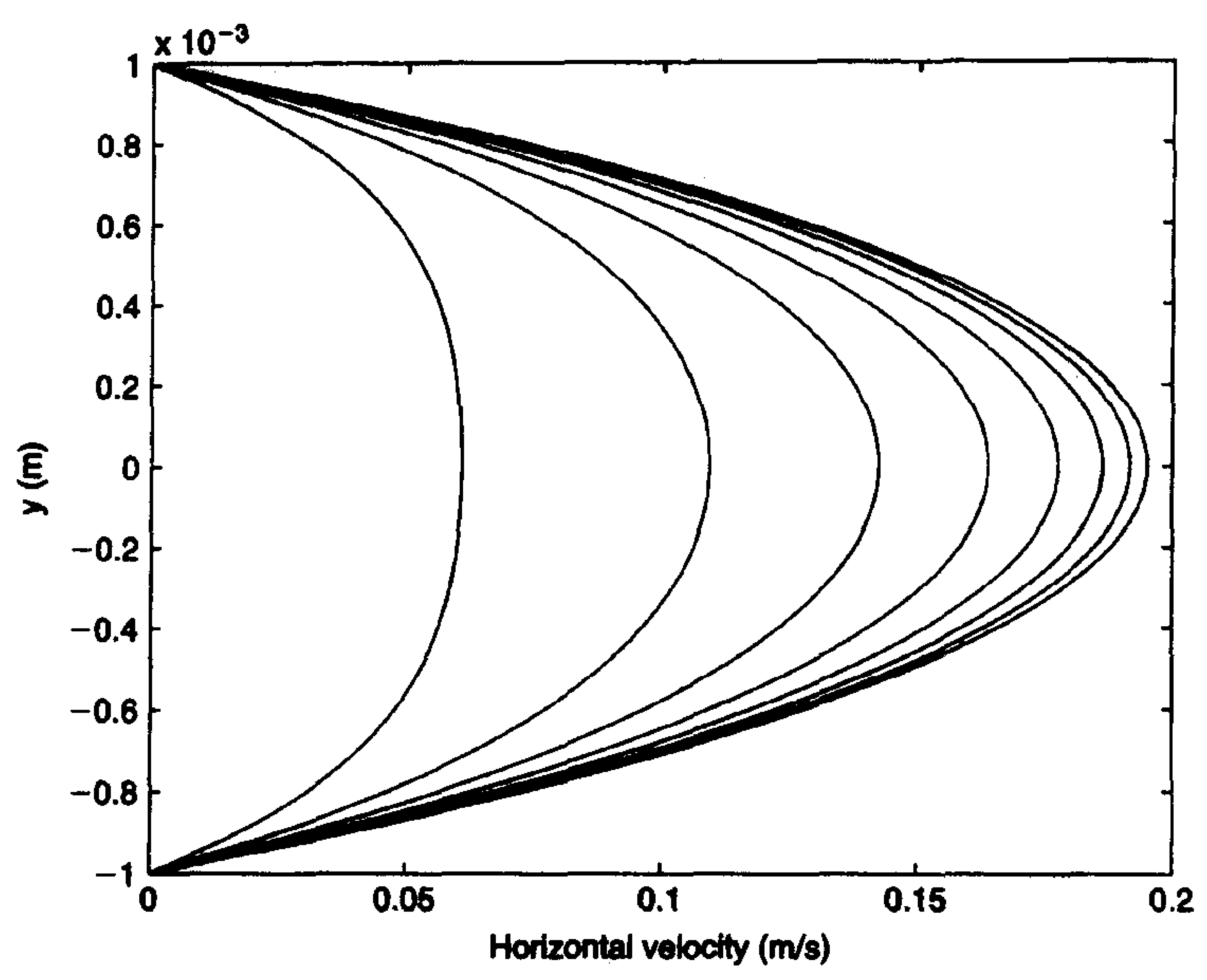


图 11.28 连接两个 0.3 cm 长平面壁的中心面轴向速度

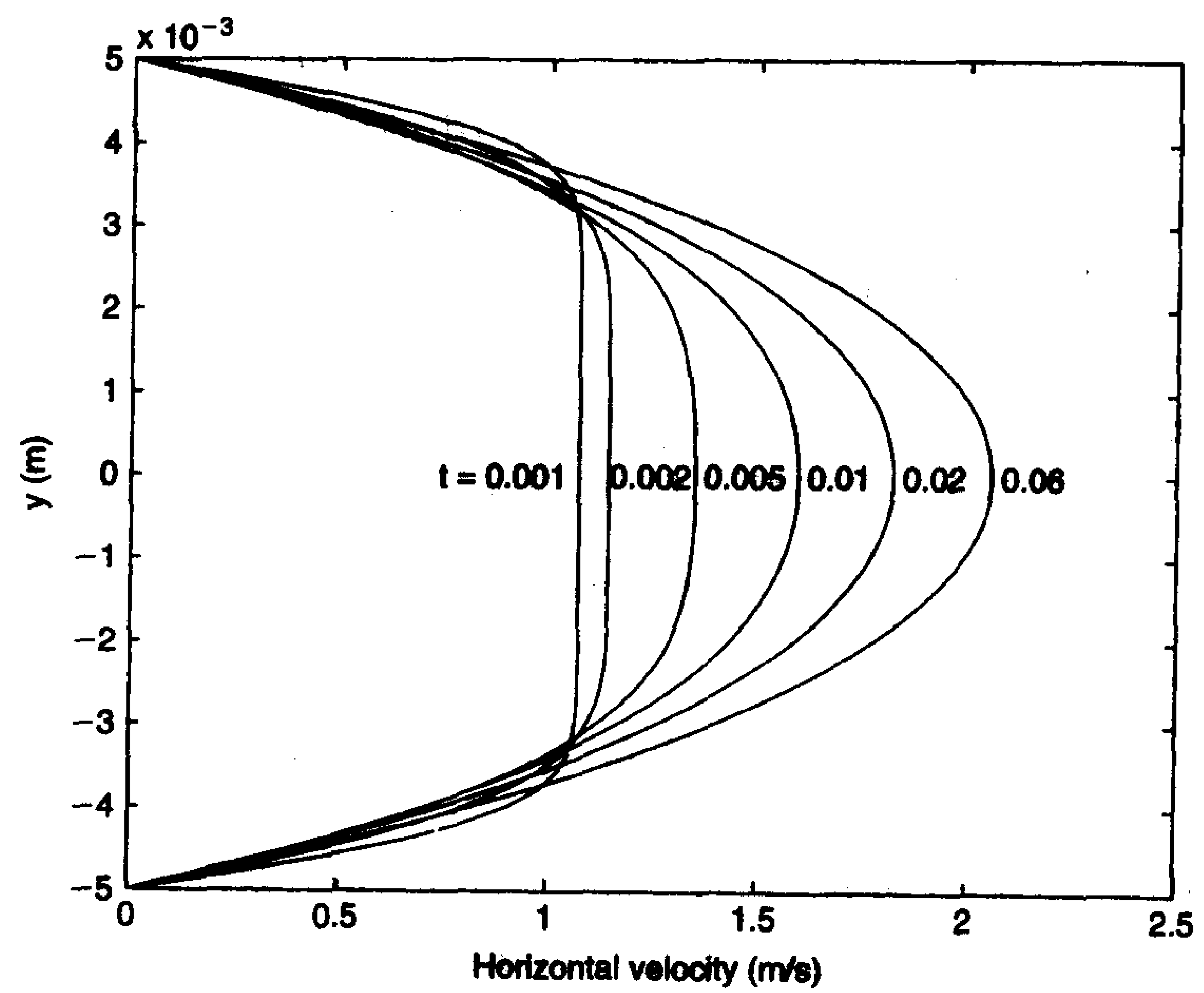


图 11.29 各个时刻通过管道中心面的轴向速度分布

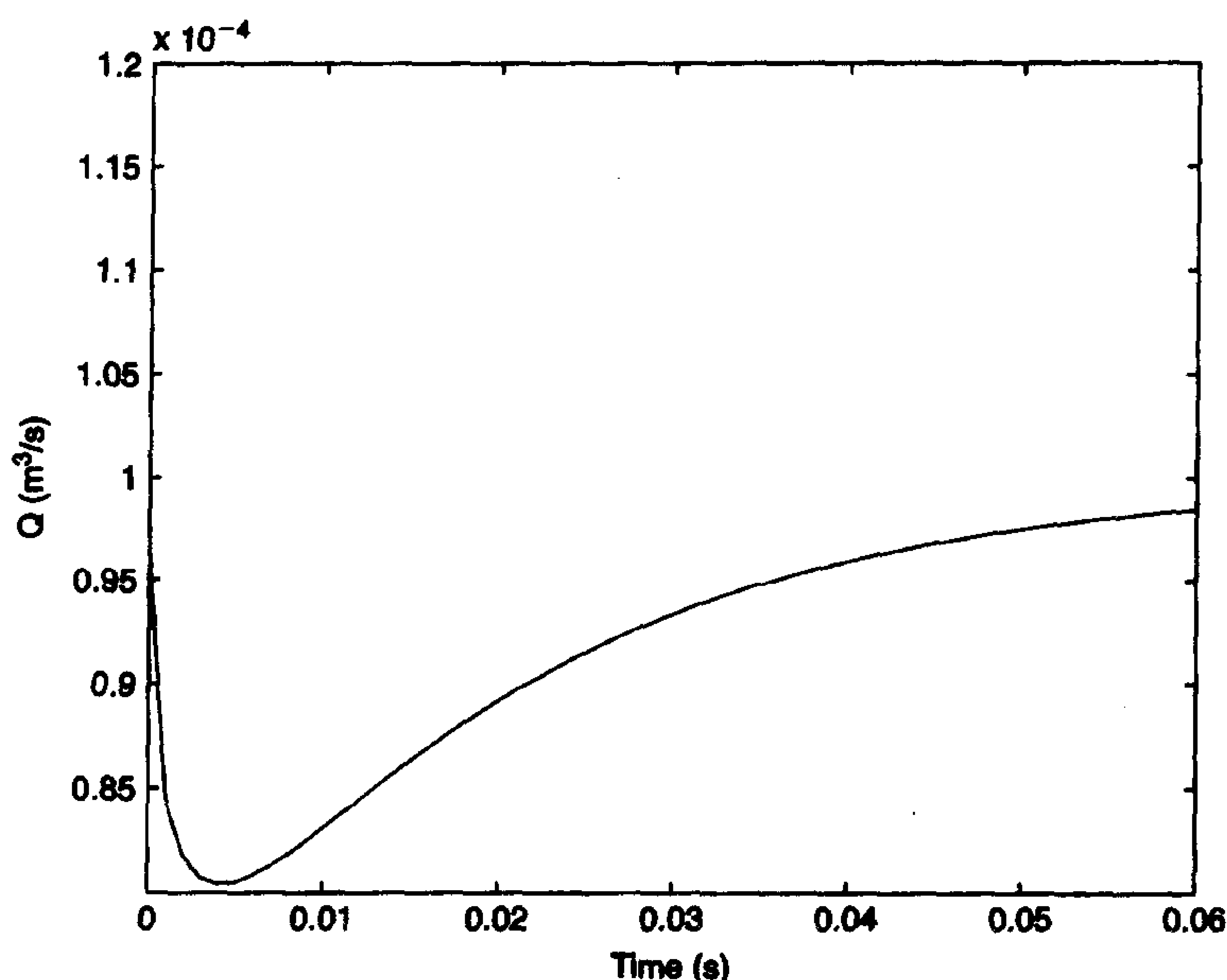


图 11.30 管道中流速与时间的关系

11.6 设定水是通过管道系统从水库 A 流向水库 B 的,如图 11.31 所示。当闸门完全打开时,流量为 0.003 m^3/s。常用的水头损失方程为:

$$\frac{P_1}{\rho g}+\frac{V_1^2}{2g}+z_1=\frac{P_2}{\rho g}+\frac{V_2^2}{2g}+z_2+\frac{\lambda L V^2}{2gD}+\sum_{m=1}^{5}K_{Lm}\frac{V^2}{2g}$$

式中,K_{Lm}是在这些区域(如图 11.31 所示)中的局部损失系数,λ 是管道的沿程损失系数,$V=4Q/\pi D^2$ 是管道中的平均速度。已知 $v_{vis}=1.3\times10^{-6}$ m^2/s 和 $\rho=1000$ kg/m^3,求管道的直径。

[答案:$D=0.046\ 98$ m。]

11.7 两个截面为矩形,面积相等的水库,由长为 L 的长管连接,高度差为 Z,如图 11.32 所示。从高度 Z 突然释放的流体产生的振动可以由下式决定①:

$$\frac{d^2Z}{dt^2}+\text{signum}(dZ/dt)p\left(\frac{dZ}{dt}\right)^2+qZ=0$$

式中,

$$p=\frac{fA_1A_2L_e}{2DaL(A_1+A_2)}\qquad q=\frac{ga(A_1+A_2)}{A_1A_2L}$$

假设液体为紊流,以使水头损失与速度平方成比例。数值 L_e 等价于合成管长度的最小损失,g 是重力加速度,f 是管子的摩擦系数,A_1 和 A_2 是两个水库的表面积,a 是管子的截面积,D 是直径。

如果 $p=0.375$ m^{-1},$q=7.4\times10^{-4}s^{-2}$,并且初始条件 $Z(0)=Z_n$ m,$dZ(0)/dt=0$ m/s,当 $Z(t_n)=0$,$Z_n=5$,10,…,50 时,试确定第一次出现的 t_n 值并绘制结果

① 参见 D.N.Roy, *Applied Fluid Mechanics*, Ellis Horwood Limited, Chichester, England, 1988 年,第 290~293 页。

曲线，曲线应如图 11.33 所示，可使用 interp1 求解 t_n。

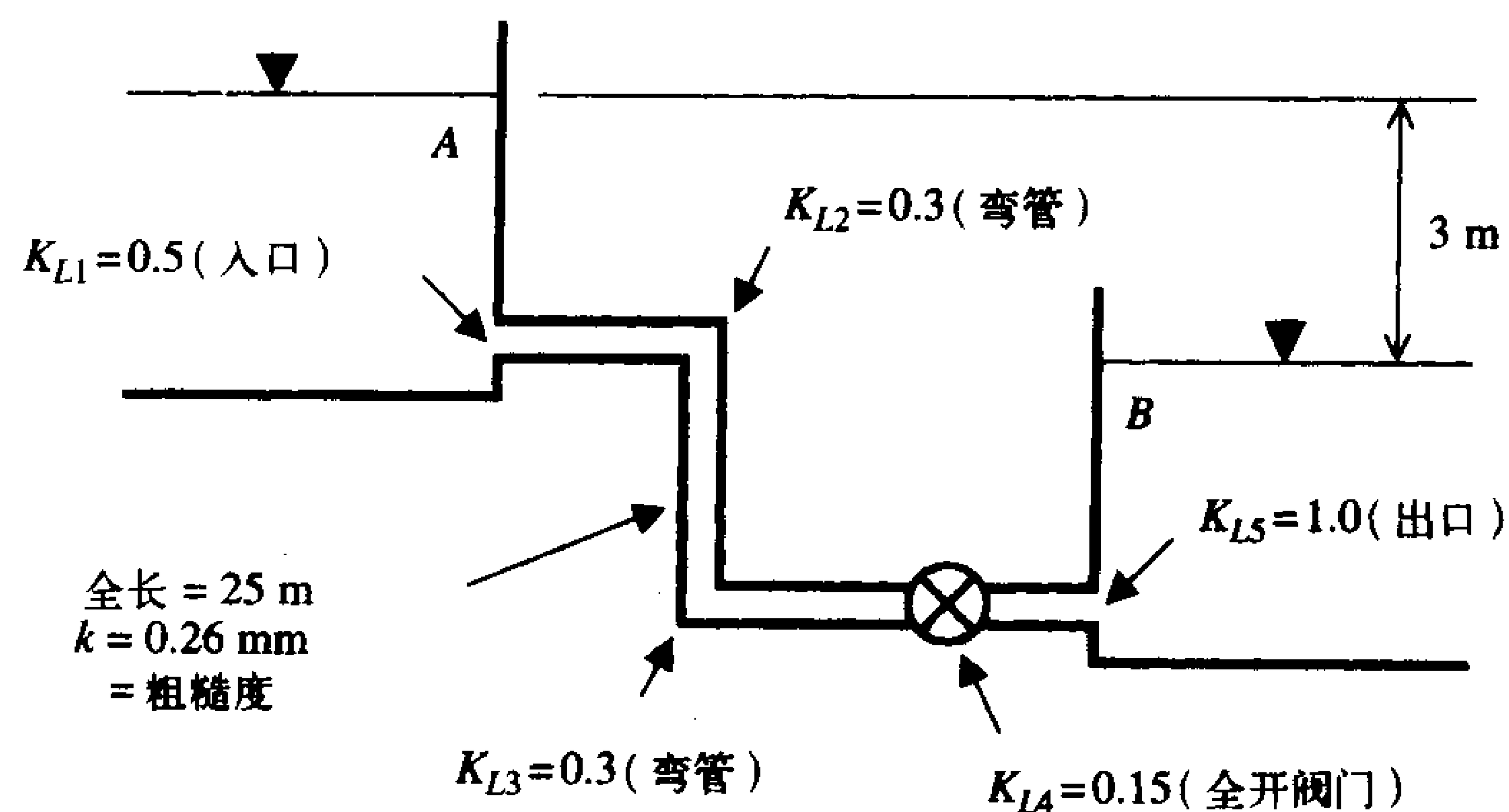


图 11.31　两水库间的管道系统

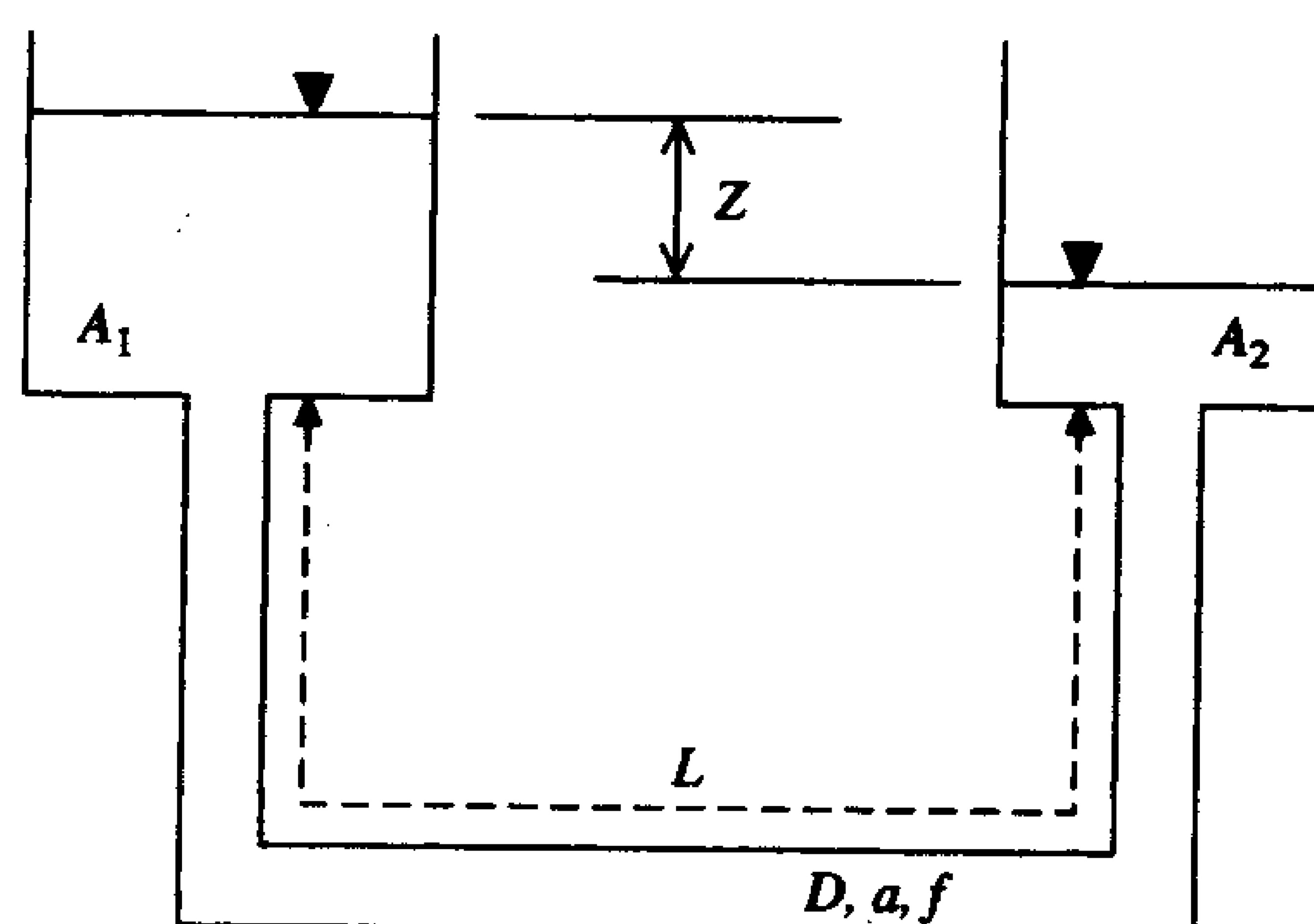


图 11.32　连通的水库

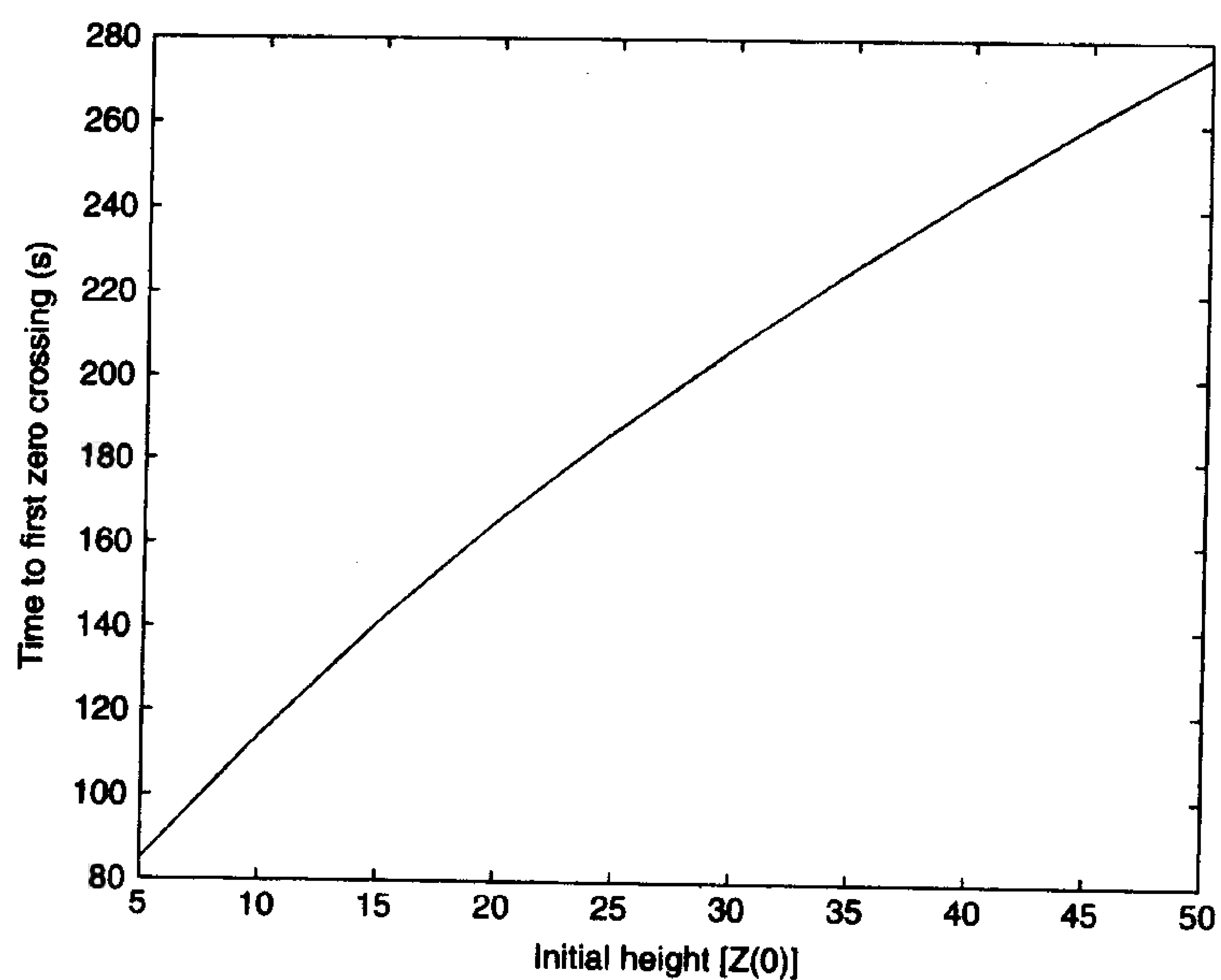


图 11.33　当 $Z(t_n) = 0$ 时第一次得到的 t_n 值与 $Z(0) = Z_n$ 的函数关系

11.8 对称薄机翼的流动可近似用势流理论表示①。翼弦在 x 轴上从 $x=0$ 延伸到 $x=c$，且由强度为 $\gamma(x)$ 的旋涡层表示。$\gamma(x)$ 为：

$$\gamma(\theta)=2\alpha V_{\infty}\frac{1+\cos\theta}{\sin\theta}$$

式中，

$$x=\frac{c}{2}(1-\cos\theta)\quad 0\leqslant\theta\leqslant\pi$$

α 是入流相对于 x 轴的冲角(单位为弧度)，V_{∞} 是流速。研究由 N 个离散旋涡近似组成的旋涡层，这个离散旋涡的间隔 $\Delta x=c/N$，强度 $\Gamma_i=\gamma(\theta_i)\Delta x$。用 11.3.3 节的第一种方法，绘出 $\alpha=10°$，$c=2$ m 和 $V_{\infty}=100$ m/s 的流线。其结果应与图 11.34 所示相同。

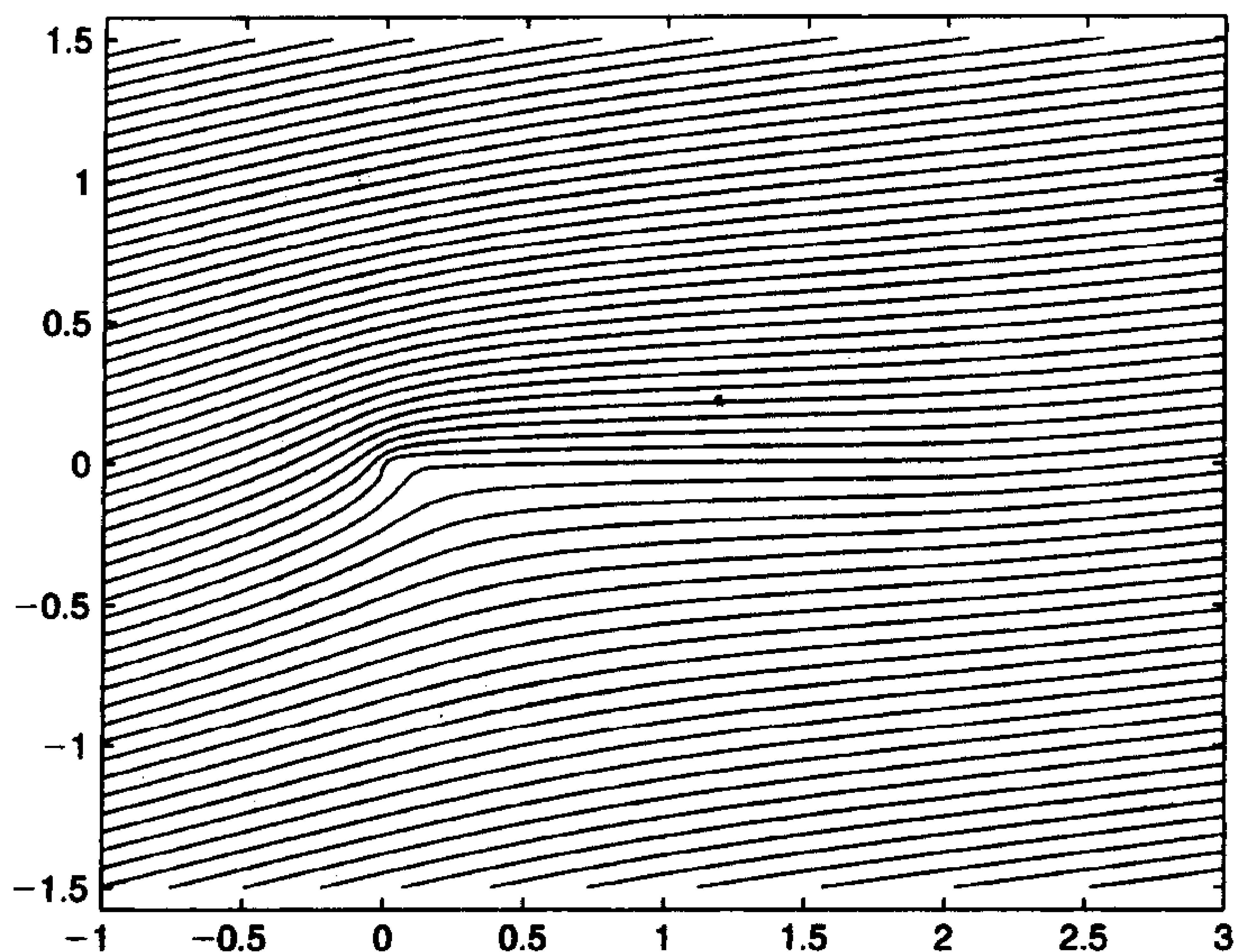


图 11.34 翼线 2 米、冲角 10°的薄机翼流线，流向从左至右，机翼从(0, 0)延伸到(2, 0)

11.9 研究管道中绕圆柱体的流场，如图 11.35 所示。假定一势流，用 `pdetool` 绘出流线。将网格和结果变量输出到 MATLAB 的命令窗口中，计算沿管道底面的速度分布。其结果应与图 11.36 所示相同。

11.10 研究通过在平面壁附近的圆柱体的势流，如图 11.37 所示。已知流以 $U=1.0$ m/s 的均匀流速通过这个圆柱体和平面壁，两偶极流的中心点坐标 (x, y) 分别为 $(0, 0.75D)$ 和 $(0, -0.75D)$，其强度是 $m=2\pi UD^2/4$，绘出流线的形状。注意环绕着每个偶极流的闭合流线并不是圆形。绘出平面壁附近的速度分布情况，然后与习题 11.9 的结果比较。

① 参见 J.D.Anderson, *Fundamentals of Aerodynamics*, McGraw-Hill, New York, NY 1991 年，第 4 章。

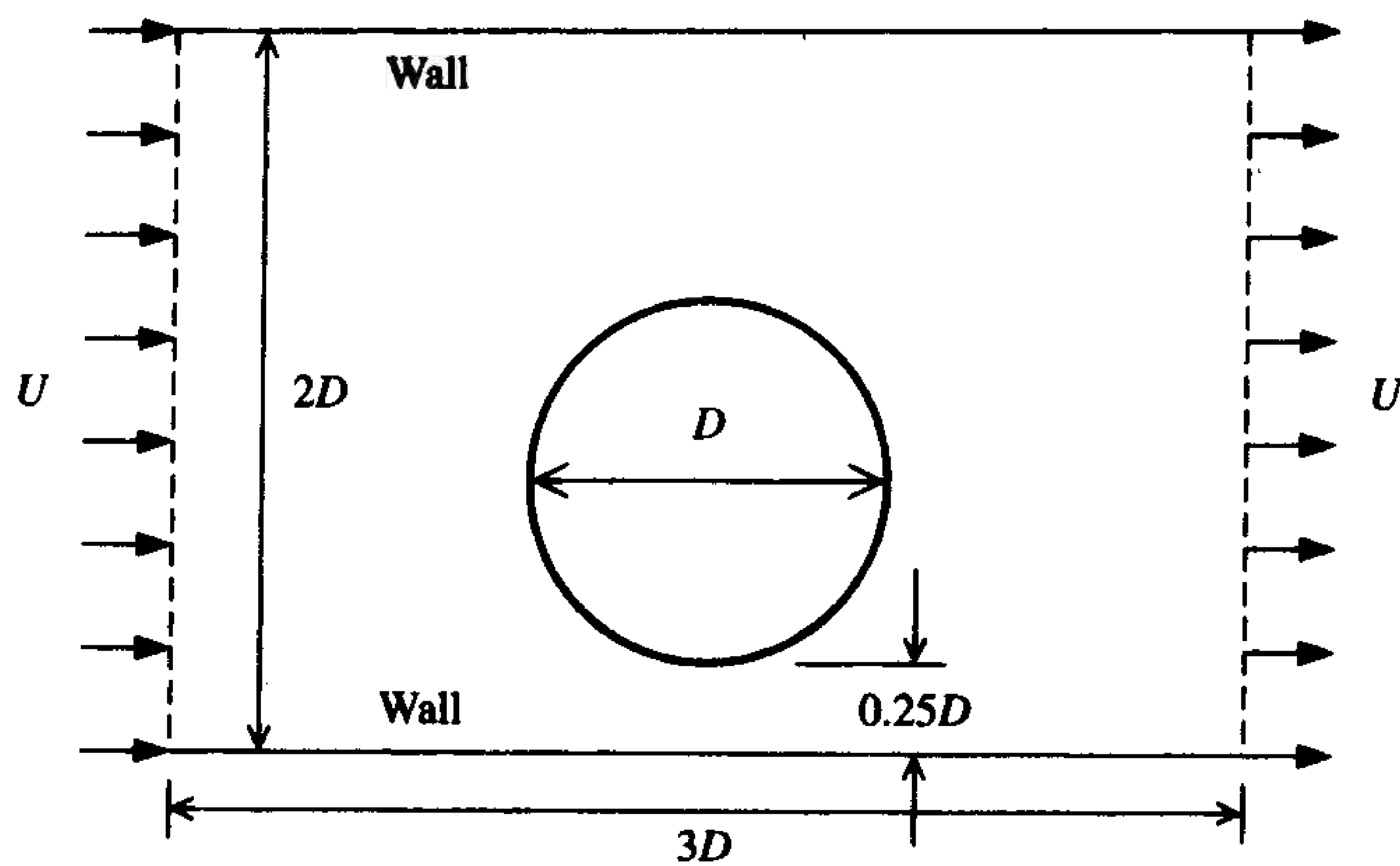


图 11.35　水流以均匀水平流速 $U = 1$ m/s，从左侧流入管道，从右侧流出管道，柱体直径 $D = 1.0$ m，中心位置坐标为 $(x, y) = (0.0, 0.75)$ m

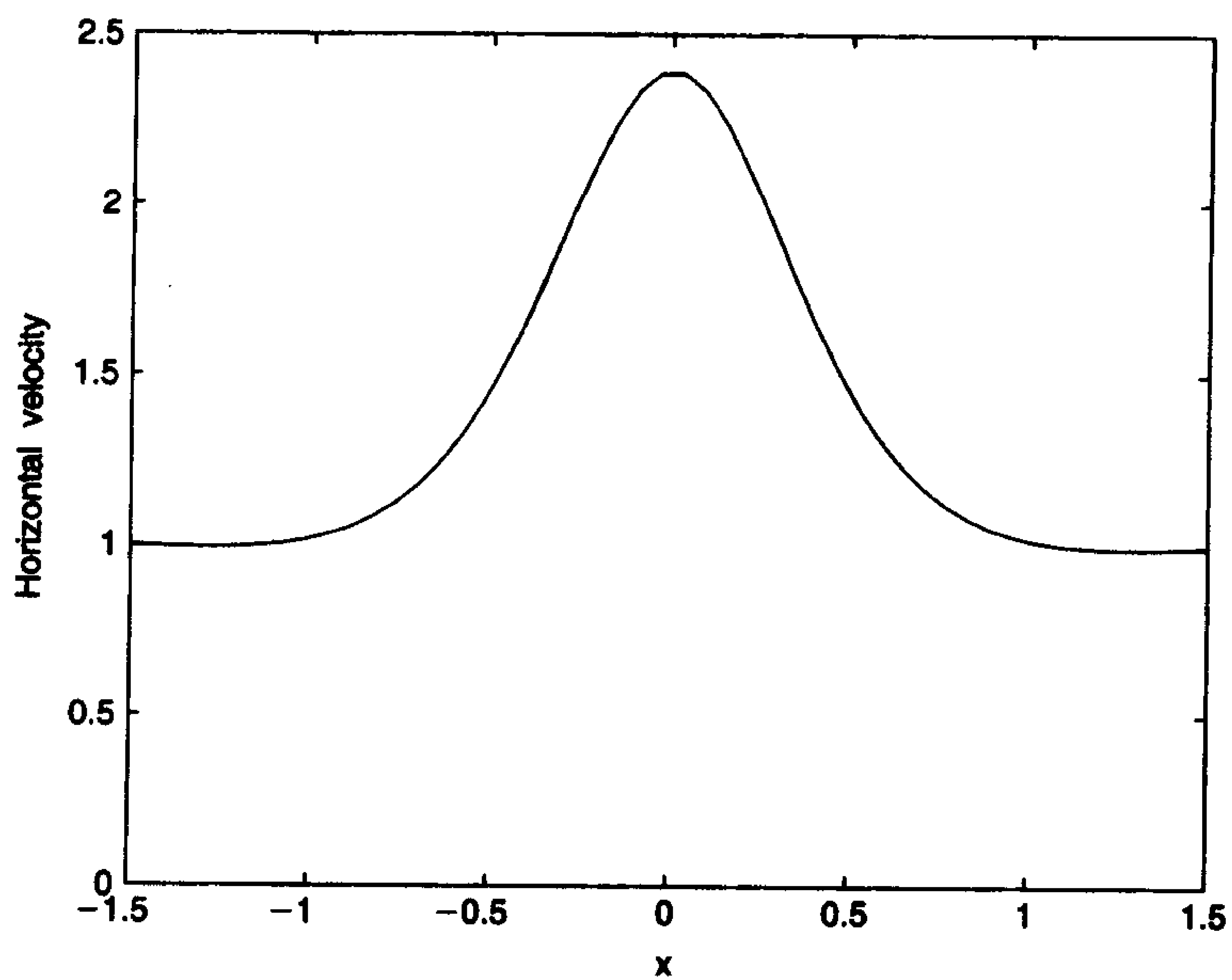


图 11.36　沿管道底面壁的水平流速

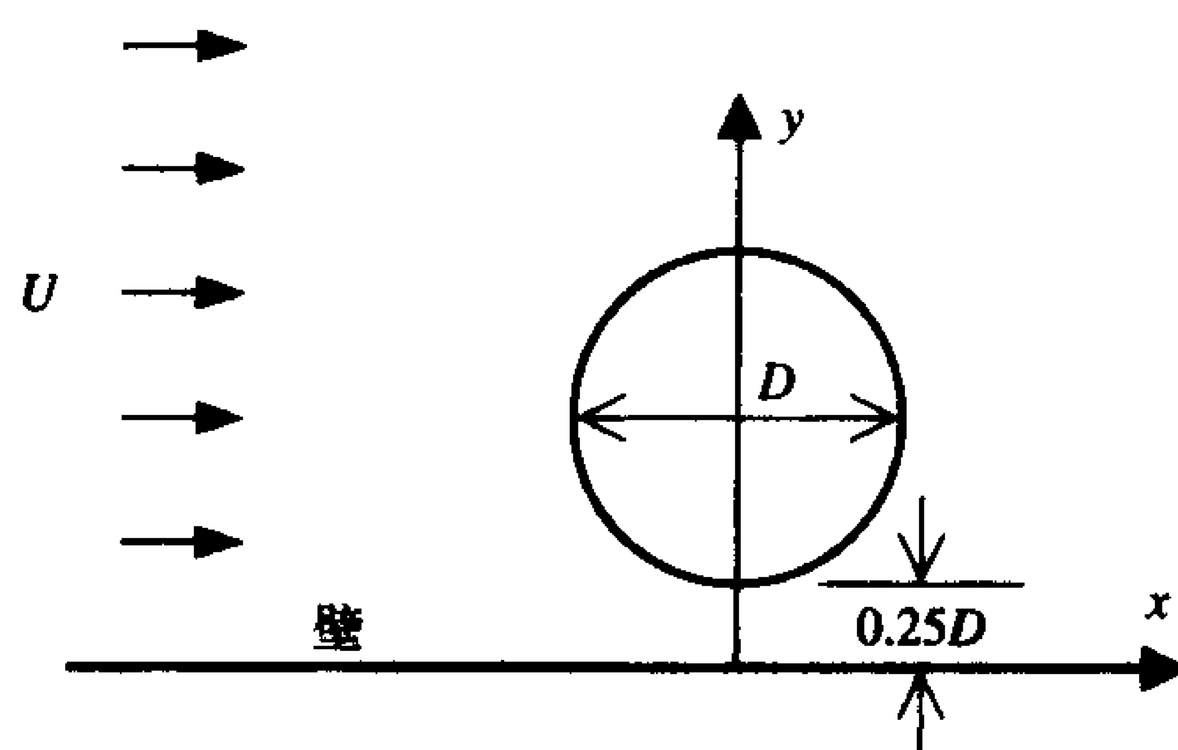


图 11.37　均匀水平流通过靠近平面壁的圆柱体

11.11 图 11.38 所示的一个薄翼型流可以用势流理论近似[①]。翼弦从 x 轴的 $x=0$ 延伸到 $x=c$,并且用沿翼弦的旋涡线来表示。旋涡线的强度 $\gamma(x)$为:

$$\gamma(\theta) = 2V_\infty\left(A_0\frac{1+\cos\theta}{\sin\theta} + \sum_{n=1}^{\infty}A_n\sin(n\theta)\right)$$

式中,

$$\frac{x}{c} = \frac{1}{2}(1-\cos\theta) \qquad 0 \leqslant \theta \leqslant \pi$$

α 是流量引入方向相对 x 轴的冲角(弧度表示),V_∞ 是流速。常数 A_n 为:

$$A_0 = \alpha - \frac{1}{\pi}\int_0^{\pi}\frac{dz}{dx}d\theta_o$$

$$A_n = \frac{2}{\pi}\int_0^{\pi}\frac{dz}{dx}\cos(n\theta_o)d\theta_o$$

其中,$z(x)$是翼弦到弧线的垂直距离。

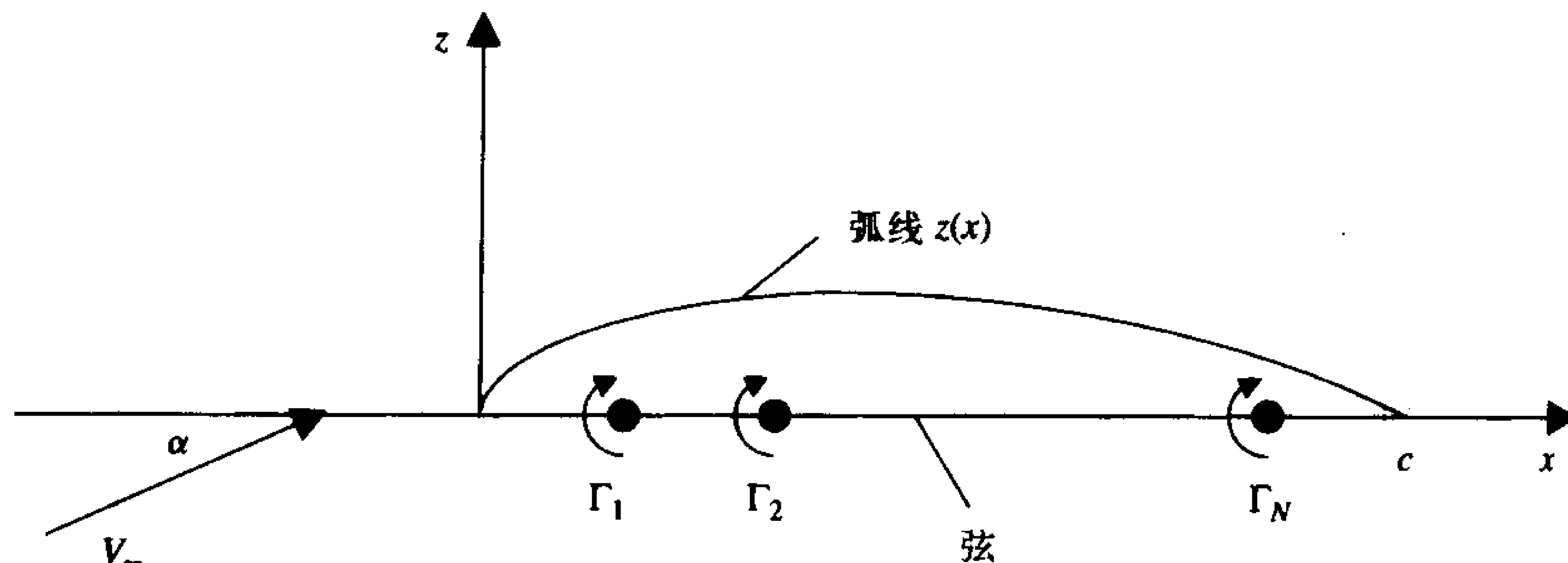


图 11.38 弦线上一个旋涡薄片的放置情况

令弧线为:

$$\frac{z}{c} = 2.6595\frac{x}{c}\left[\left(\frac{x}{c}\right)^2 - 0.6075\frac{x}{c} + 0.1147\right] \quad 0 \leqslant x/c \leqslant 0.2025$$

$$= 0.022\,08\left(1-\frac{x}{c}\right) \quad 0.2025 \leqslant x/c \leqslant 1.0$$

其中,距离 z 是对翼弦的法向距离。当 $n = 0, 1, \cdots, 20$ 时,求 A_n 的值。在 $0 \leqslant x/c \leqslant 1$范围内,令沿着 x 轴的旋涡线近似用一系列离散的旋涡值 N_v 来表示(参见 11.3.3 节)。旋涡由距离 $\Delta x = c/N$ 分割,拥有的强度 $\Gamma_i = \gamma(\theta_i)\Delta x$。使用 contour 绘制出 $\alpha=4°$, $c=2$ m 和 $V_\infty = 100$ m/s 时的流线图。曲线图中的拱形线应该接近流线。

[部分答案:$A_0=0.0412$, $A_1=0.0955$, $A_2=0.0792$, $A_3=0.0568$。]

11.12 在 z 平面上,Joukowski 翼型从前缘到后缘沿 x 轴方向的直线距离称为弦长。上下翼面沿 y 轴方向最大垂直距离定义为翼型厚度 t,如图 11.39 所示。弦线(x 轴)与翼型中线的最大距离定义为翼型的拱长 b。(翼型中线定义为上下翼面的一半。)

① 参见 J.D.Anderson, 出处同上。

t/L 是 ξ_{off} 的函数，b/L 是 η_{off} 的函数。半径从 $-0.2R$ 到 $0R$ 时有 $\eta_{off}/R = [0.01, 0.02, 0.03, 0.04]$，绘出 t/L 与 ξ_{off} 的函数关系。半径从 $0.0R$ 到 $0.04R$ 时有 $\xi_{off}/R = [-0.2, -0.15, -0.10, -0.05, 0]$，绘出 b/L 与 η_{off} 的函数关系。其结果应与图 11.40所示相同。

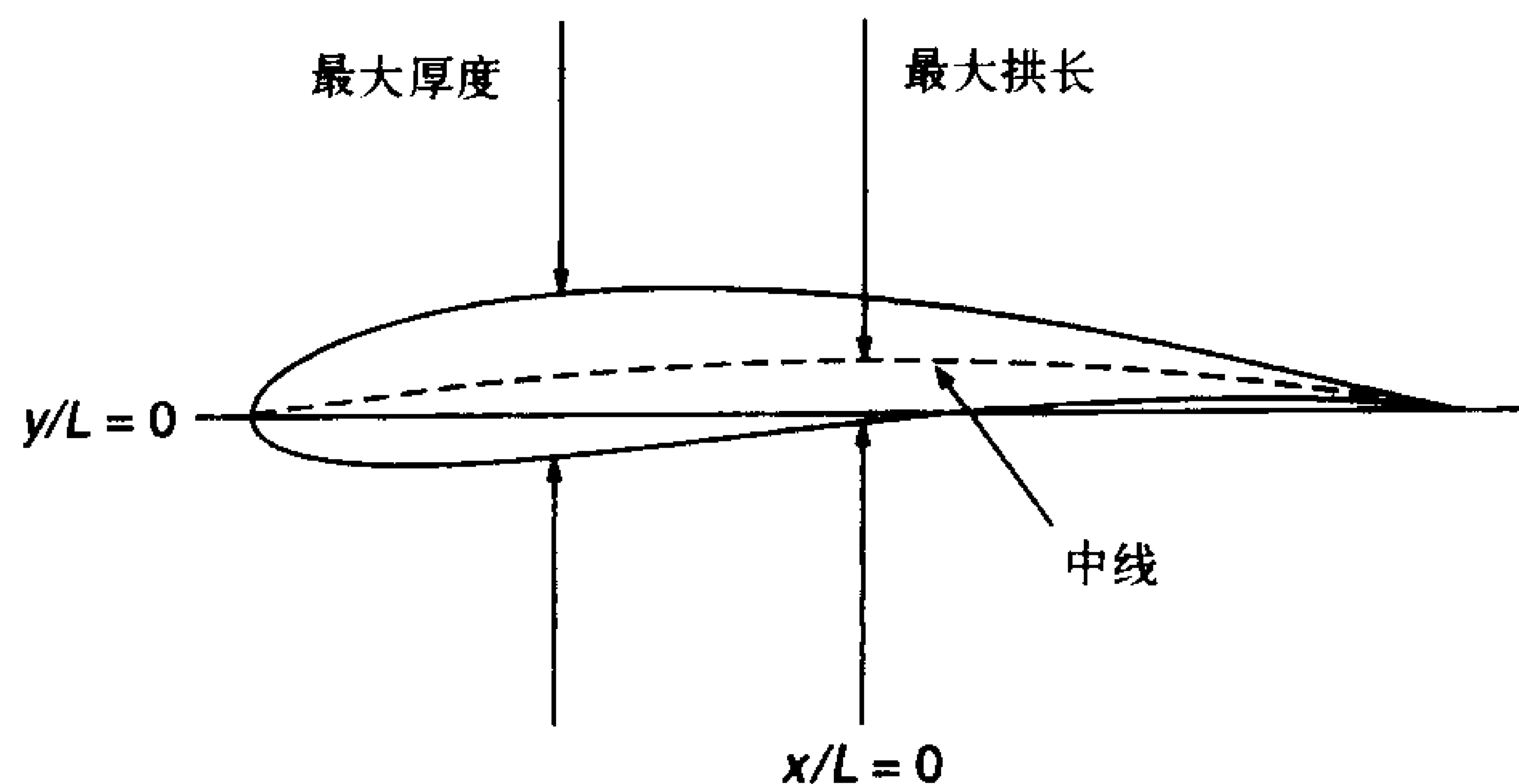


图 11.39　最大翼型厚度和最大翼型拱长的定义

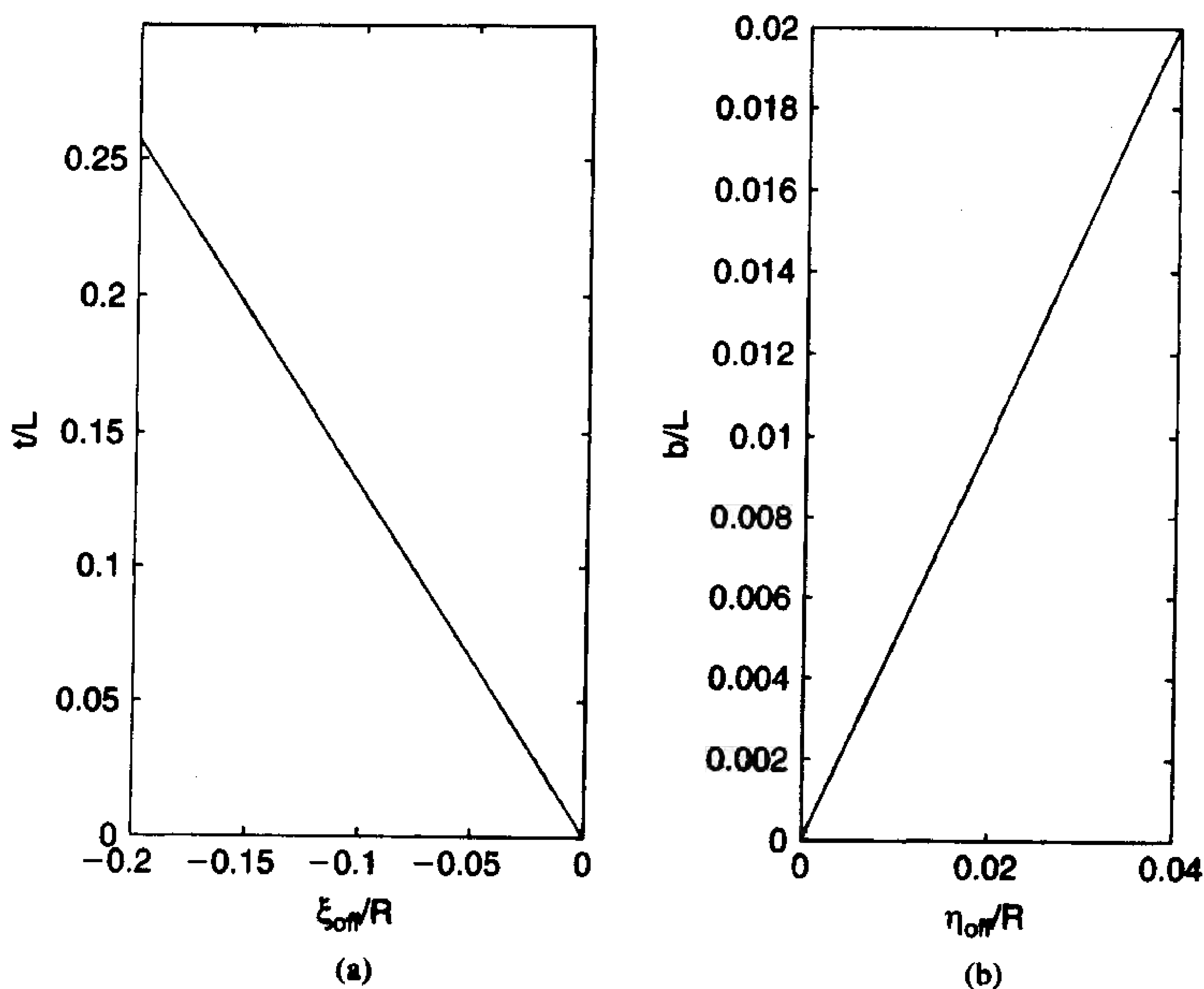


图 11.40　(a)翼型厚度与 ξ_{off} 的函数关系；(b)翼型拱长与 η_{off} 的函数关系。在图(a)中绘制了对应 $\eta_{off}/R = 0, 0.01, 0.02, 0.03$ 和 0.04 的五条曲线。在图(b)中绘制了对应 $\xi_{off}/R = -0.2, -0.15, -0.10, -0.05$ 和 0.0 的五条曲线。注意在每幅图中五条曲线彼此首尾相连

11.13 已知 $R = 1.0$，$\alpha = 6°$，$\xi_{off} = -0.093R$，$\eta_{off} = 0.08R$（即 $t/L = 0.1215$，$b/L = 0.0401$）。绘出 Joukowski 翼型的上下翼面上压力系数的分布[参见式(11.24)]。其结果应与图 11.41 所示相同。

11.14 翼型升力系数由下式给出：

$$C_L = \frac{F_L}{\rho Q^2 L/2}$$

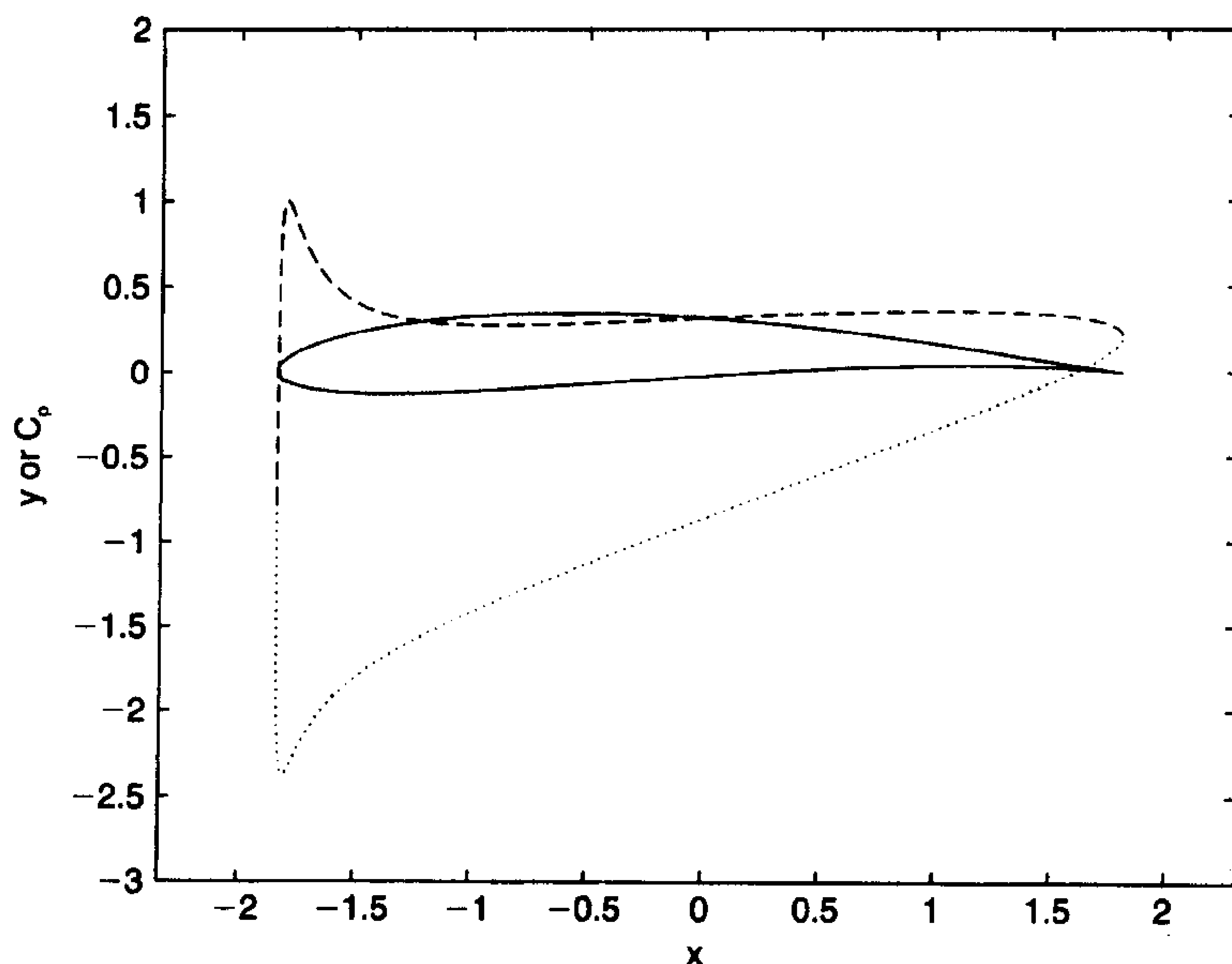


图 11.41　已知 $R=1.0$, $Q=1.0$, $\alpha=6°$, $\xi_{off}=-0.093R$, $\eta_{off}=0.08R$($t/L=0.1215$, $b/L=0.0401$)时,翼型和在上翼面(点线表示)及下翼面(虚线表示)的压力系数分布图

式中,$F_L=\rho Q\Gamma$ 是单位拱度的升力。绘制 $\xi_{off}=-0.093R$ 和 $\eta_{off}=[0,0.02R,0.04R,0.06R,0.08R]$时 C_L 与冲角 α(从 −10°到 10°)的函数关系。其结果应与图 11.42所示相同。为了便于计算,令 $R=1$。

11.15 当堰顶和水库中自由液面的垂直距离 $y_R=10.0$ ft 时,并且,用 $n=0.014$, $b=5.0$ ft 及 $m=1.0$ 来描述长棱柱形槽,找出水库流量与槽坡面之间的函数关系。其结果应与图 11.43 所示相同。

11.16 研究已知条件为 $y_R=10.0$ ft, $n=0.014$, $b=5.0$ ft 及 $S_0=0.001$ 时从水库到棱柱形槽的流量。绘制流量和槽侧壁坡面 m 的函数关系。其结果应与图 11.44 所示相同。

11.17 研究已知条件为 $y_R=10.0$ ft, $n=0.014$, $b=5.0$ ft, $m=1.0$ 及 $S_0=0.003$ 时从水库到棱柱形槽的流量。注意,从图 11.43 可知,S_0 在临界坡面上流量达到最大值,即 1666 ft^3/s。假定槽足够长以致流动可以逐渐达到急流深度,计算槽内水面高度的轮廓线。其结果应与图 11.45 所示相同。

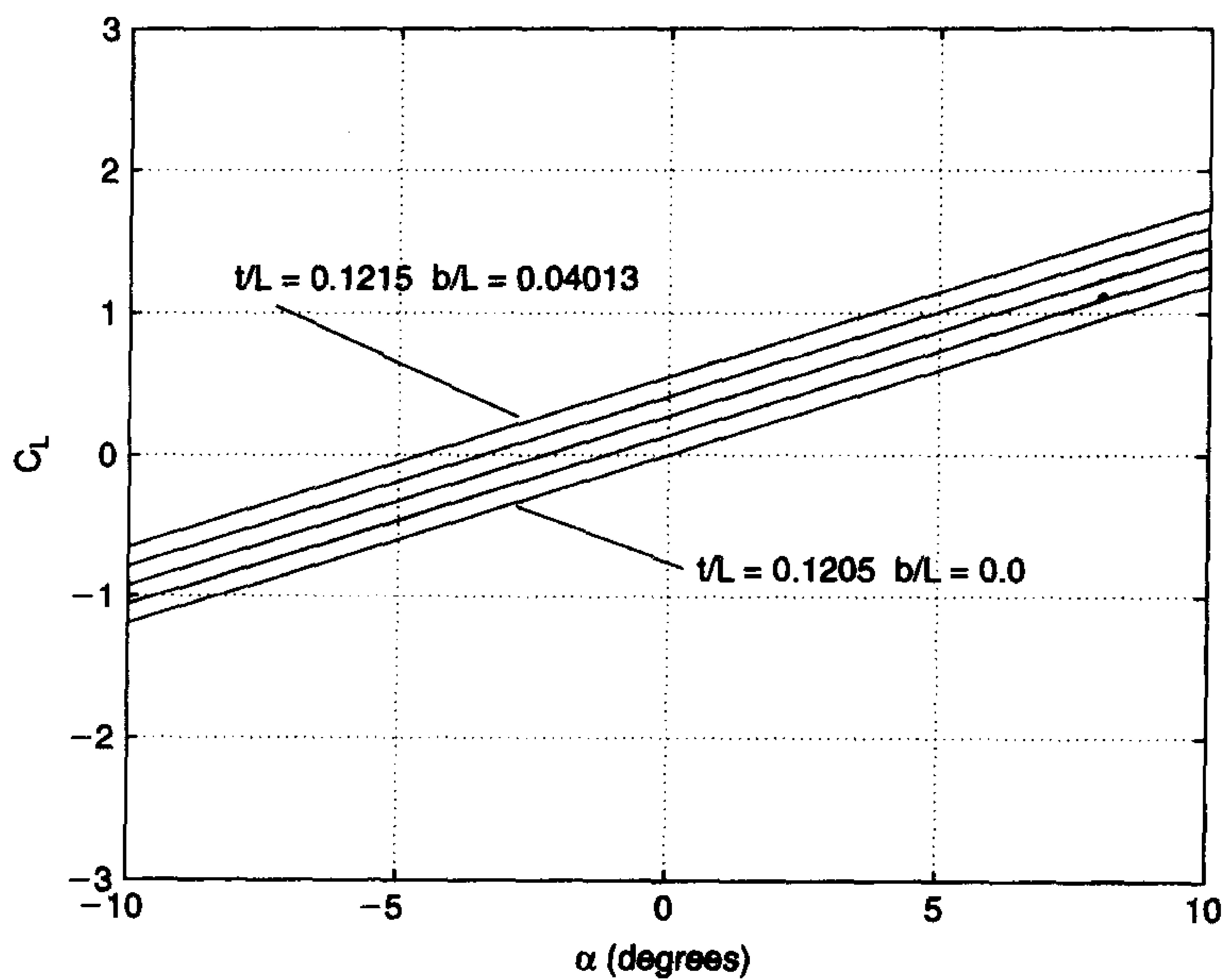


图 11.42　已知 $\xi_{off} = -0.093R$ 和 $\eta_{off} = 0, 0.02R, 0.04R, 0.06R$ 及 $0.08R$ 时升力系数 C_L 与冲角 α 的函数关系

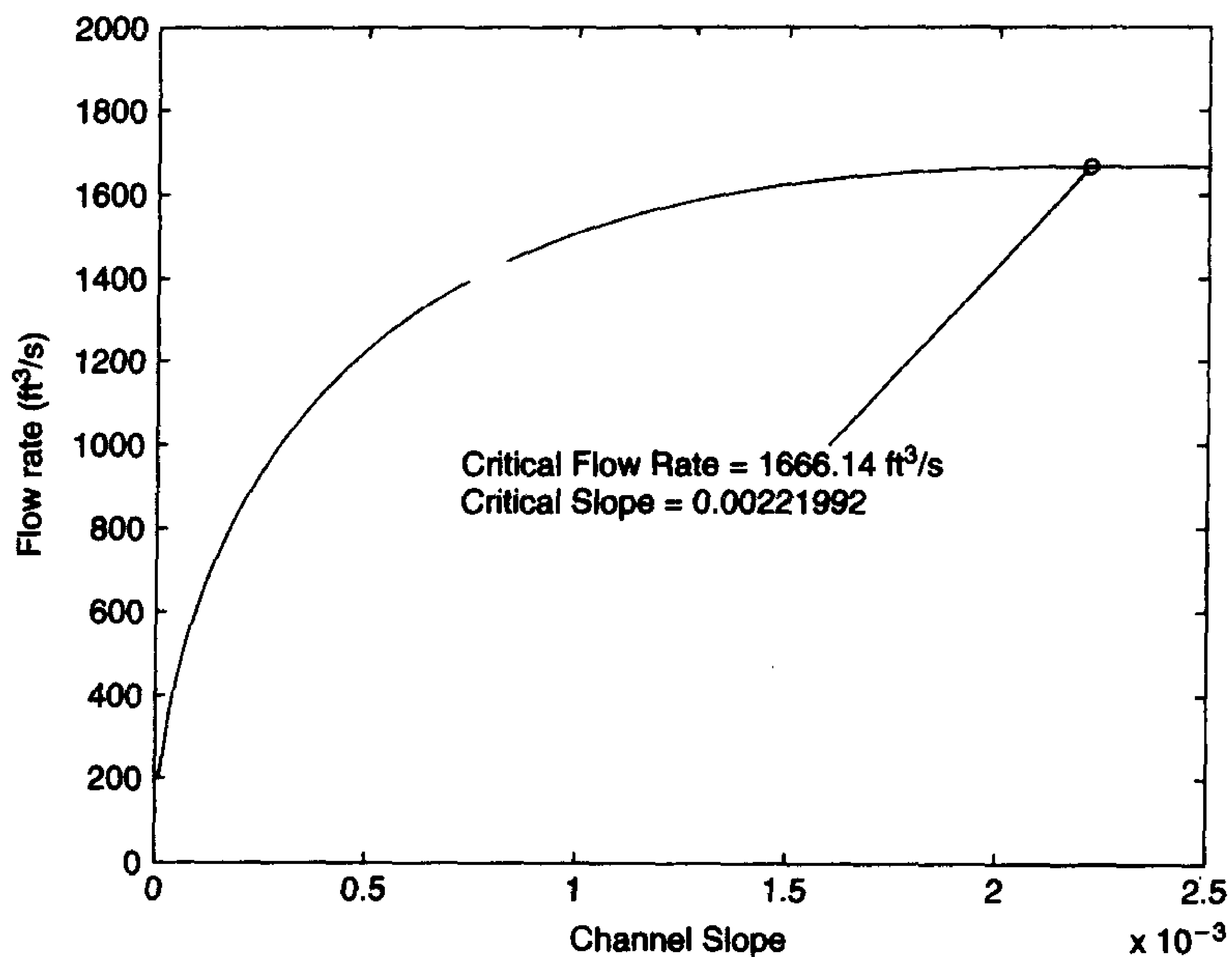


图 11.43　已知 y_R = 10.0 ft, n = 0.014, b = 5.0 ft 及 m = 1.0 时水库流量问题的流量与槽坡面的函数关系

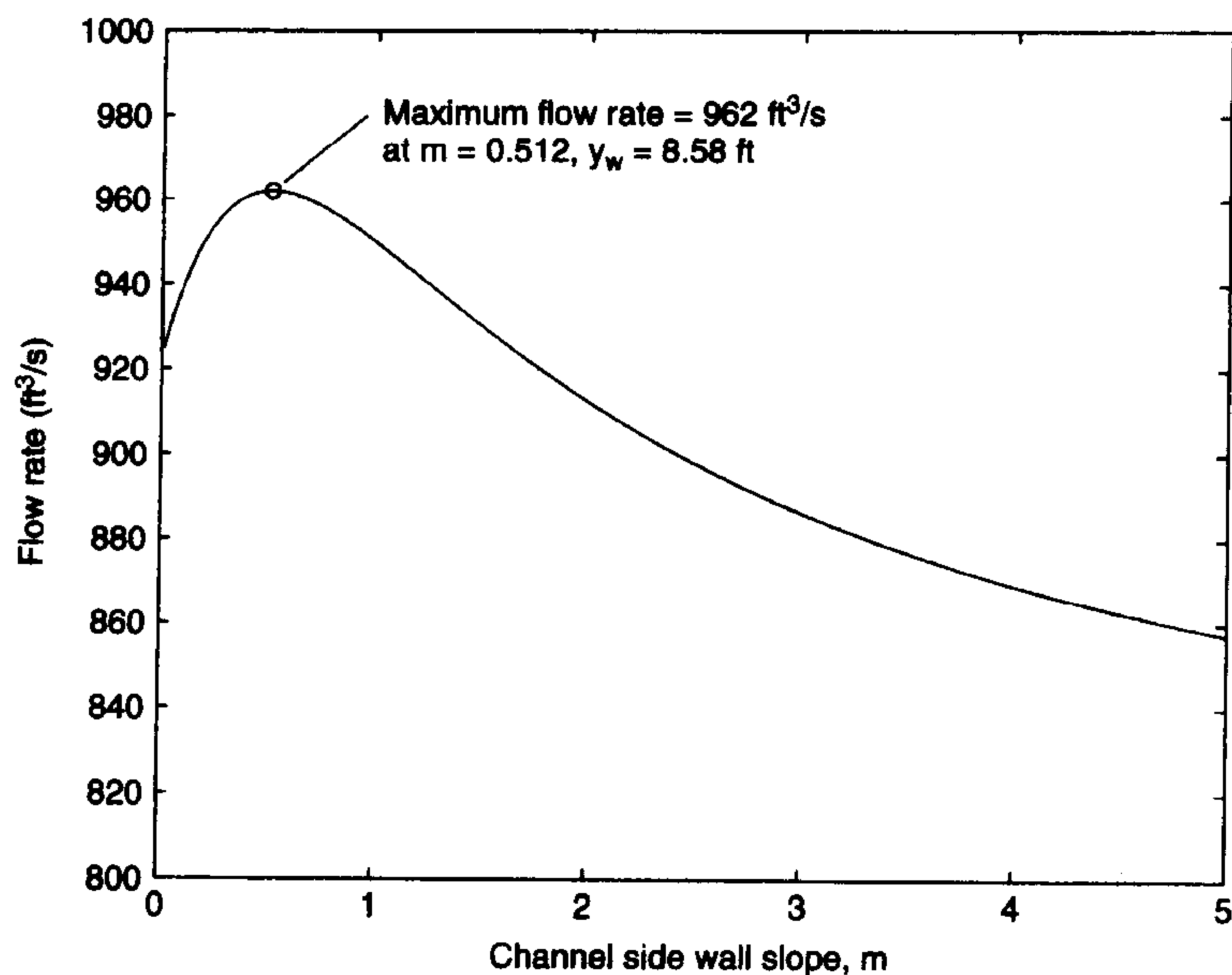


图 11.44 已知 y_R = 10.0 ft, n = 0.014, b = 5.0 ft 及 S_0 = 0.001时流量与槽侧壁坡面 m 的函数关系

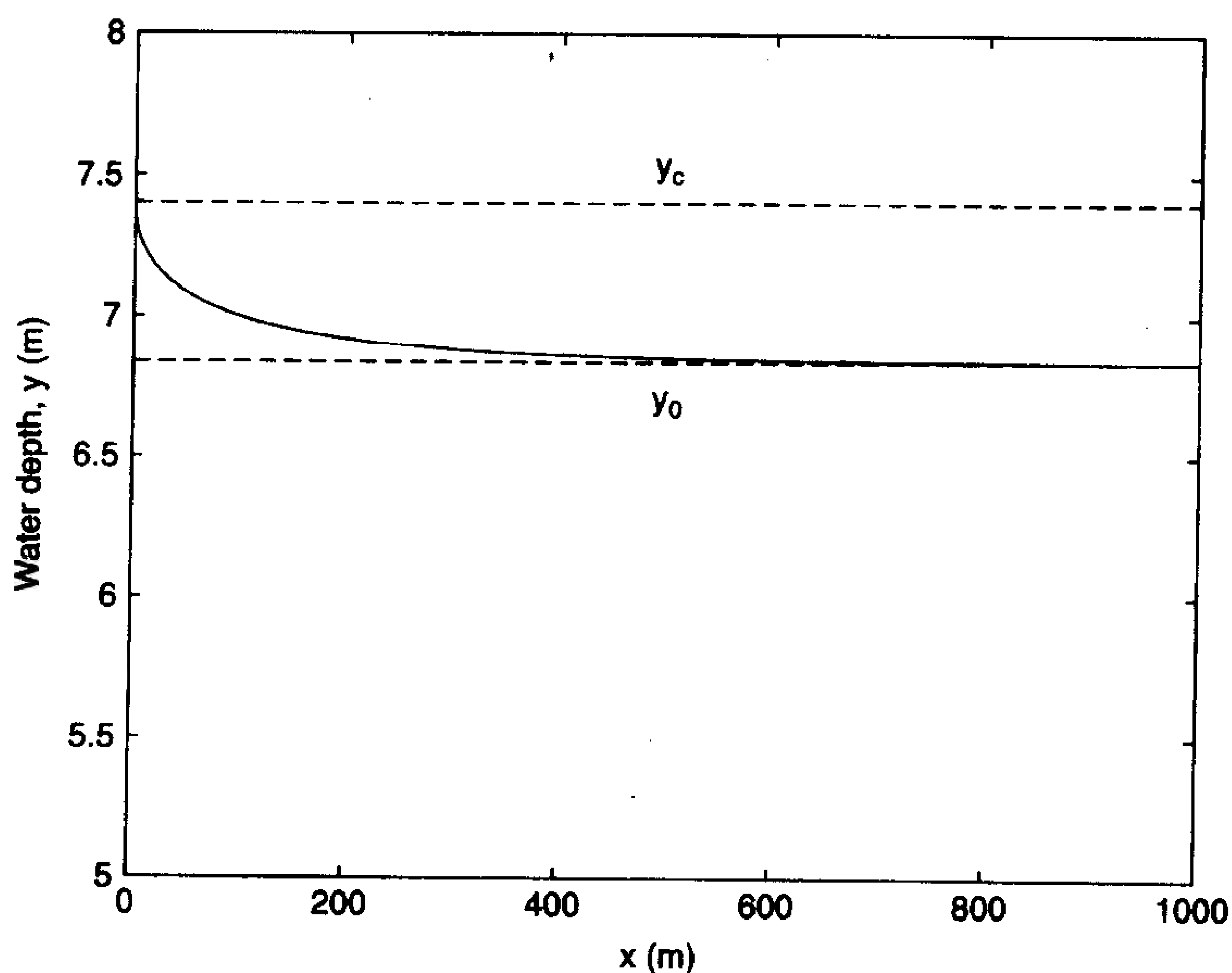


图 11.45 在 $x=0$ 和在 $n=0.014$, b = 5.0 ft, m = 1.0 及 $S_0=0.003$ 时产生急流坡面的临界流出口条件下的槽内表面轮廓线

参考文献

1. B.R.Munson, D.F.Young, and T.H.Okiishi, *Fundamentals of Fluid Mechanics*, 3rd ed., John

Wiley & Sons, New York, NY, 1998.

2. J.D. Anderson, *Fundamentals of Aerodynamics*, 2nd ed., McGraw-Hill, New York, NY, 1991.
3. V.L. Streeter, E.B. Wylie, and K.W. Bedford, *Fluid Mechanics*, 9th ed., McGraw-Hill, New York, NY, 1998.
4. R.W. Fox and A. McDonald, *Introduction to Fluid Mechanics*, 5th ed., John Wiley & Sons, New York, NY, 1998.

第12章 热 传 导

以下给出分析和设计导热、对流和辐射传热的几种方法。

12.1 导热

12.1.1 具有表面对流的半无限大平板的瞬态导热

半无限大固体平板初始温度分布均匀,边界表面 $\eta = 0$ 处具有对流,该平板瞬时温度分布由下式给出[①]:

$$\theta(\eta,\tau) = \mathrm{erfc}\left[\frac{\eta}{2\tau}\right] - \exp[\eta + \tau^2]\,\mathrm{erfc}\left[\frac{\eta}{2\tau} + \tau\right]$$

式中,erfc 是修正误差函数:

$$\theta(\eta,\tau) = \frac{T(\eta,\tau) - T_\infty}{T_\infty - T_i}$$

$$\tau = \frac{h}{k}\sqrt{\alpha t}$$

$$\eta = \frac{hx}{k}$$

x 是空间坐标,t 是时间,h 是传热系数,k 是固体的热导,T_∞是环境温度,α 是固体热扩散系数。

例 12.1 半无限大固体的瞬态导热和温度分布

用两种方法绘制半无限大固体中的温度。在第一种方法中,在 $0 \leqslant \eta \leqslant 5$ 和 $0.01 \leqslant \tau \leqslant 3$ 范围内,绘出温度分布曲线,结果如图 12.1 所示。在第二种方法中,将温度表示成 τ 的函数。其中,$0.01 \leqslant \tau \leqslant 4$,且 η 取 0, 1, …, 5 六个不同的值。绘出温度分布曲线,结果如图 12.2 所示。程序如下:

```
tau = linspace(0.01,3,30);eta = linspace(0,5,20);
[x,t] = meshgrid(eta,tau);
theta = inline('erfc(0.5 * x./t) - exp(x + t.^2). * erfc(0.5 * x./t + t)','x','t');
figure(1)
mesh(x,t,theta(x,t))
xlabel('\eta')
ylabel('\tau')
zlabel('\theta')
figure(2)
eta = linspace(0,5,6);
tau = linspace(0.01,4,40);
```

① 参见 F.P.Incropera 和 D.P.DeWitt, *Fundamentals of Heat and Mass Transfer*, 第四版, John Wiley & Sons, New York, NY, 1996 年, 第 239 页。

```
for k = 1:6
  thet = theta(eta(k),tau);
  plot(tau,thet)
  text(.92*4,1.02*thet(end),['\eta='num2str(eta(k))])
  hold on
end
xlabel('\eta')
ylabel('\theta')
```

执行程序后,得到图 12.1 和图 12.2。

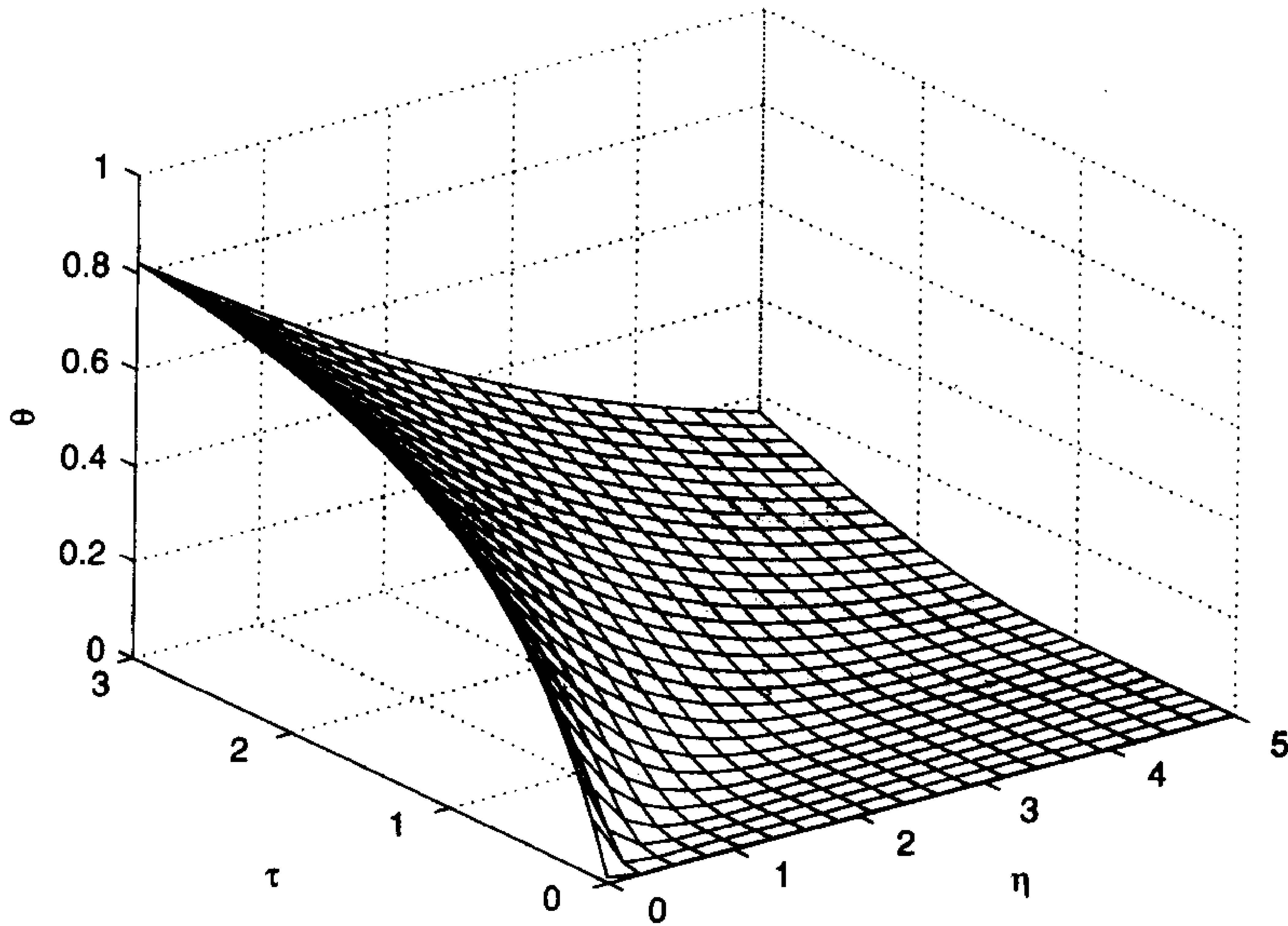


图 12.1　半无限大固体温度随位置 η 和时间 τ 变化的格状曲面图

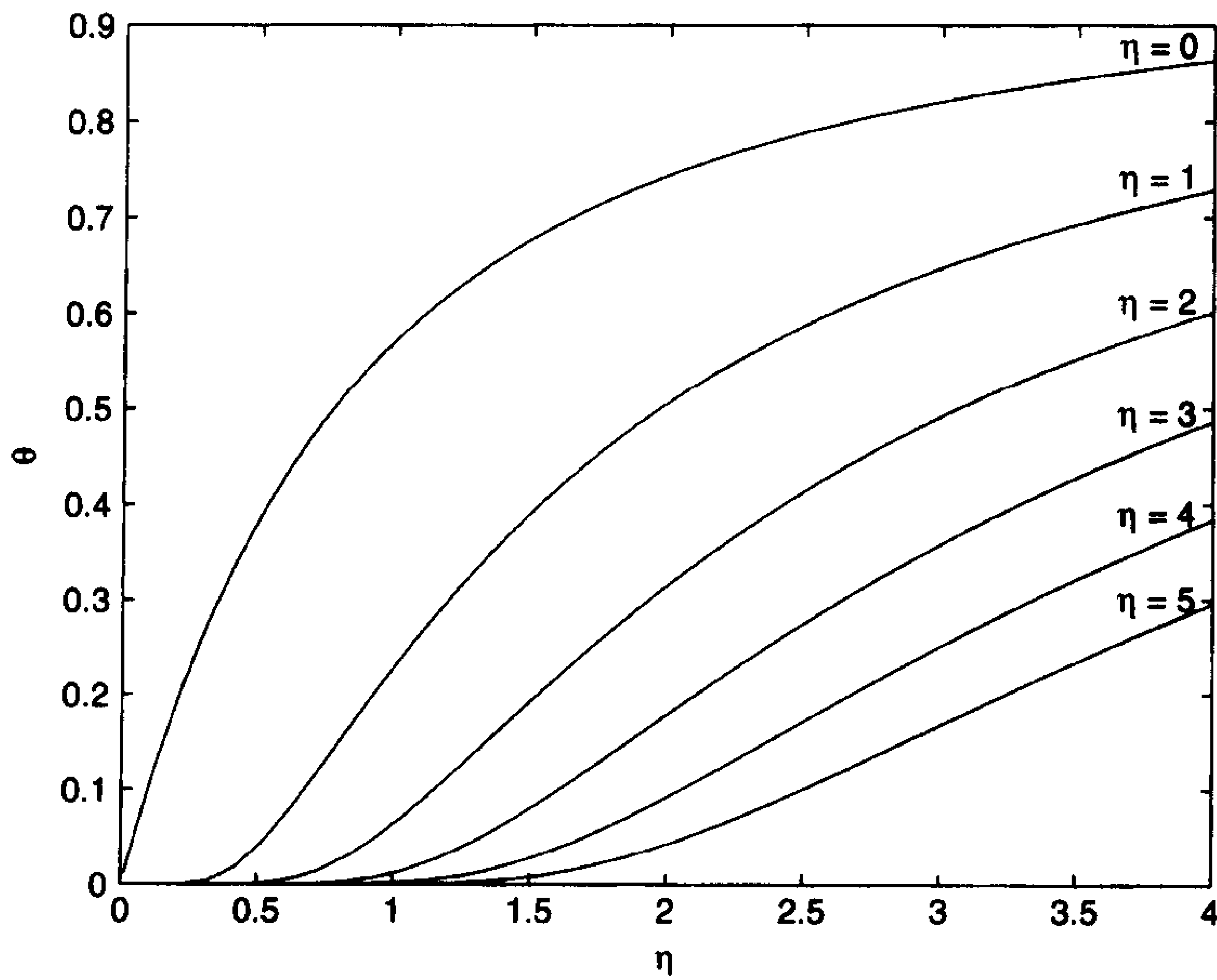


图 12.2　半无限大固体温度随位置 η 和时间 τ 变化的曲线

12.1.2 具有对流的无限长圆柱体的瞬态导热

一个无限长圆柱固体初始温度均匀分布,表面具有对流,其瞬时温度分布由下式给出[①]:

$$\theta(\xi,\tau) = \sum_{n=1}^{\infty} C_n \exp(-\zeta_n^2 \tau) J_0(\zeta_n \xi)$$

其中:

$$\theta(\xi,\tau) = \frac{T(\xi,\tau) - T_\infty}{T(\xi,0) - T_\infty}$$

而且,

$$C_n = \frac{2}{\zeta_n} \frac{J_1(\zeta_n)}{J_0^2(\zeta_n) + J_1^2(\zeta_n)}$$

$J_m(x)$是一类 m 阶贝塞尔函数,$\tau = \alpha t/a^2$, α 是热扩散系数,a 是圆柱体的半径,t 是时间,$\xi = r/a$,r 是圆柱体的径向距离,T_∞是环境温度,ζ_n 是下式的正根:

$$\frac{J_1(\zeta_n)}{J_0(\zeta_n)} - \frac{Bi}{\zeta_n} = 0$$

式中,$Bi = ha/k$ 是毕奥特数,h 是传热系数,k 是圆柱体的热导。

例 12.2 具有对流的无限长圆柱体的瞬态导热

在区间 $0 \leqslant \xi \leqslant 1$ 和 $0 \leqslant \tau \leqslant 1.5$ 内且 $Bi = 0.5$ 时,用 ζ_n 的最小 15 个正根绘出 $\theta(\xi, \tau)$的分布图。程序如下:

```
Bi = 0.5; nroots = 15; r = zeros(1, nroots);
guess = 0.01;
CylinderRoots = inline('x. * besselj(1,x) - Bi * besselj(0,x)','x','Bi');
options = optimset('Display','off');
for k = 1:nroots
  r(k) = fzero(CylinderRoots,[guess guess + 1.1 * pi],options,Bi);
  guess = 1.05 * r(k);
end
tau = linspace(0,1.5,20);
[t,rt] = meshgrid(tau,r);
Fn = exp( - t. * rt.^2);
cn = 2 * besselj(1,r)./(r. * (besselj(0,r).^2 + besselj(1,r).^2));
ccn = meshgrid(cn,tau);
pro = ccn'. * Fn;
rstar = linspace(0,1,20);
[R,rx] = meshgrid(rstar,r);
Jo = besselj(0,rx. * R);
the = Jo' * pro;
[rr,tt] = meshgrid(rstar,tau);
mesh(rr,tt,the')
xlabel('\xi')
ylabel('\tau')
zlabel('\theta')
```

---

① 参见 F.P.Incropera 和 D.P.DeWitt,出处同上,第 229 页。

```
view(49.5, -34)
```

程序中值 *guess* 由 **CylinderRoots** 的图形确定。程序执行结果参见图 12.3。

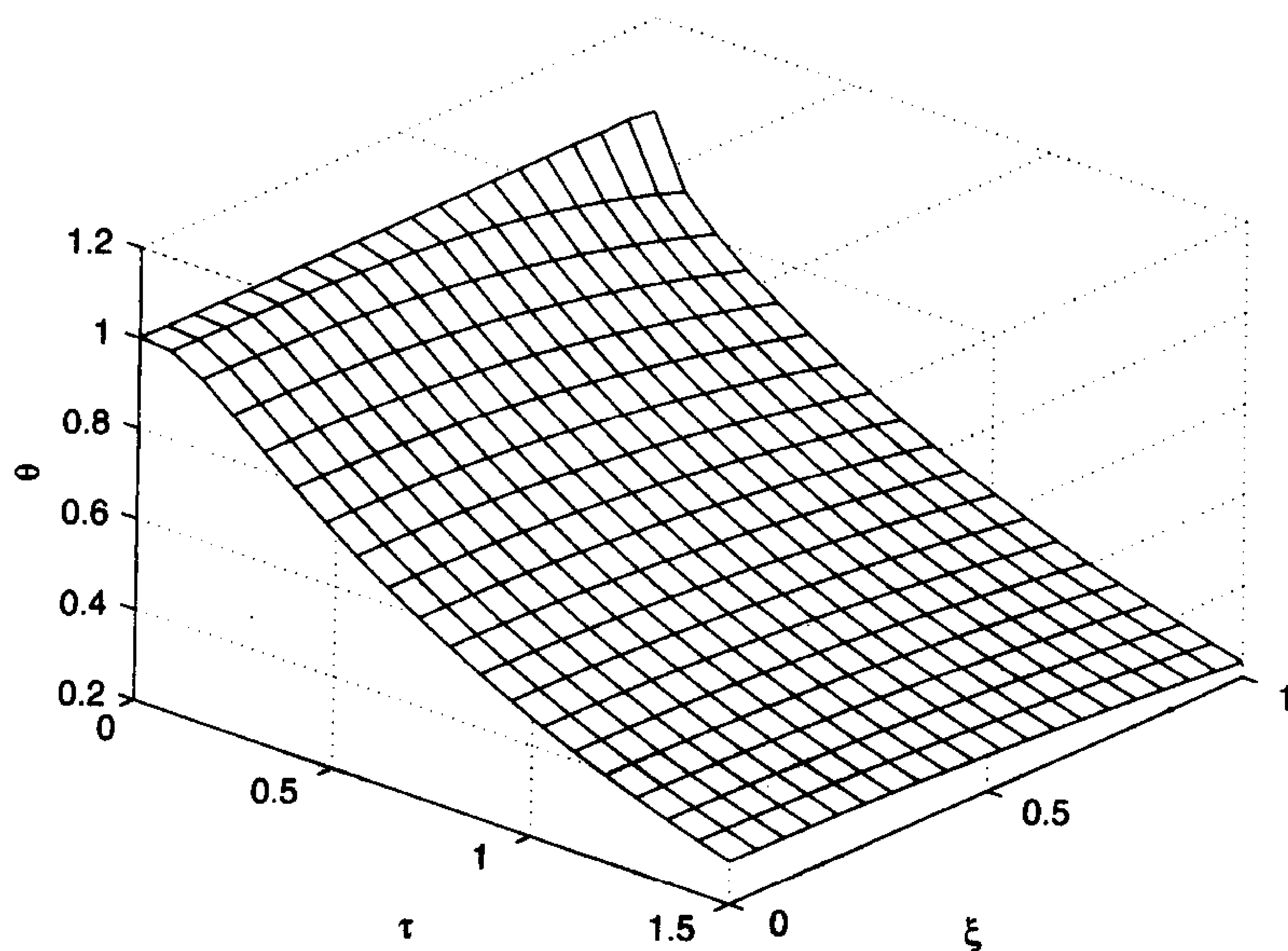


图 12.3　圆柱体内温度分布是时间 τ 和径向位置 ξ 的函数

12.1.3　有热源的一维瞬态导热

一维瞬态导热的特征方程为：

$$\frac{1}{\alpha}\frac{\mathrm{d}T}{\mathrm{d}t}=\frac{\mathrm{d}^2T}{\mathrm{d}x^2}+\frac{q}{k}$$

其中，T 是温度，t 是时间，x 是空间坐标，α 是热扩散系数，k 是热导系数，q 是体积热源。通过引入无量纲变量，将上式转化成无量纲形式：

$$\xi=\frac{x}{L}\qquad \tau=\frac{\alpha t}{L^2}\qquad Bi=\frac{hL}{k}$$

$$\theta=\frac{T-T_\infty}{T_i-T_\infty}\qquad \sum=\frac{L^2q}{k(T_i-T_\infty)}\qquad \chi=\frac{-q''L}{k(T_i-T_\infty)}$$

其中，L 是特征长度，T_i 代表初始温度，q''是热流量，T_∞ 为对流边界的流体温度。在非对流边界条件下，T_∞ 根据需要取值，但其值不能等于 T_i。特征方程利用这些变量变为：

$$\frac{\partial\theta}{\partial\tau}=\frac{\partial^2\theta}{\partial\xi^2}+\sum \tag{12.1}$$

区域如图 12.4 所示。区域两端典型的边界条件有：

恒温：

$$\theta=\theta_w$$

恒定热流量：

$$\frac{\partial\theta}{\partial\xi}=\chi_w$$

对流:

$$\frac{\partial \theta}{\partial \xi} = -Bi\theta_w$$

式中,负号是在对流条件下对左边界进行修正,下标 w 代表壁的值。

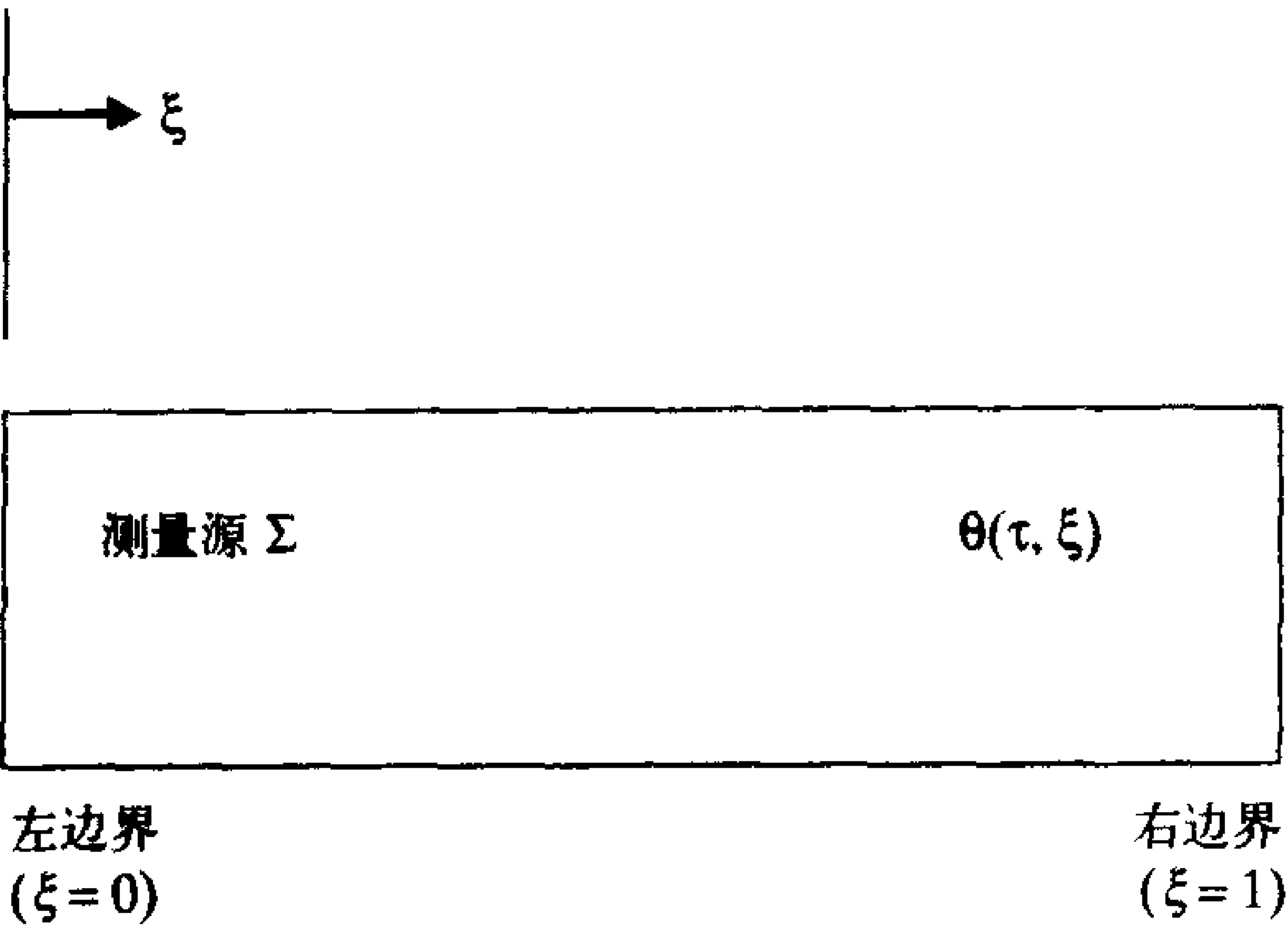


图 12.4 有源Σ的一维瞬态导热的几何特性

式(12.1)是一维抛物线偏微分方程。一种解法是变量分离法,适用于边界条件受限的情况。12.1.2 节的结果就是用这种方法得到的。偏微分方程(PDE, Partial Differential Equation)工具箱提供了另一种解法。由于 PDE 不适用于只有一个空间变量的情况,所以解决这类问题比较麻烦。可以解只有一个空间变量的抛物线偏微分方程,函数 pdepe 提供了这种能力。以下说明使用 pdepe 解决这个问题的方法。

例 12.3 有热源的一维瞬态导热

研究表 12.1,表中描述了来自瞬态热源的一维系统的数据。在固体内的五个位置:$\xi = 0.0,\ 0.25,\ 0.5,\ 0.75$ 和 1.0,将无量纲的温度确定为无量纲时间的函数。为求得解,注意到 pdepe 需要三个函数:第一个是 **pdetran**,用于定义偏微分方程;第二个是 **pdeIC**,用于定义初始条件;第三个是 **pdeBC**,用于定义边界条件。这些函数的语法可以从 pdepe 的在线帮助中获得。主函数和子函数的输入如表 12.1 所示:

表 12.1 获得图 12.5 的输入值

参数	值
边界条件和源	
无量纲热源强度,Σ	1
左边界毕奥特数	0.1
右边界无量纲温度 $\theta(1)$	0.55
初始条件	
左边界 $\theta_i(0)$	1
$\theta_i(0)=1$ 和 $\theta_i(1)=0.55$ 间的线性分布	
有限差分参数	
在 $0<\xi<1$ 内等间隔网格点数	21
等间隔时间步长数	101
积分范围(归一化时间)	1

```
function Transient1D
Bi = 0.1;Tr = 0.55;Sigma = 1;L = 21;Lt = 101;
```

```
xi = linspace(0,1,L);
tau = linspace(0,1,Lt);
options = odeset;
sol = pdepe(0,@pdetran1D;@pdeIC,@pdeBC,xi,tau,options,Bi,Tr,Sigma);
theta = sol(:,:,1);
g = [1,6,11,16,L];
for k = 1:5
  plot(tau,theta(:,g(k)),'k-')
  hold on
  if k == 1
     text (0.5,1.02 * theta(Lt,g(k),' \ xi = 0.0 and 0.25')
  elseif k > 2
     text(0.5,1.04 * theta(Lt,g(k),[' \ xi = 'num2str(xi(g(k)))])
  end
end
axis([0 1 0.5 1])
xlabel(' \ tau')
ylabel(' \ theta')
function[c,f,s] = pdetran1D(x,t,u,DuDx,Bi,Tr,Sigma)
c = 1;
f = DuDx;
s = Sigma;
function T0 = pdeIC(x,Bi,Tr,Sigma)
T0 = 1-0.45 * x;
function[pl,ql,pr,qr] = pdeBC(xl,ul,xr,ur,t,Bi,Tr,Sigma)
pr = ur-Tr;
qr = 0;
pl = -Bi * ul;
ql = 1;
```

程序执行结果参见图 12.5。

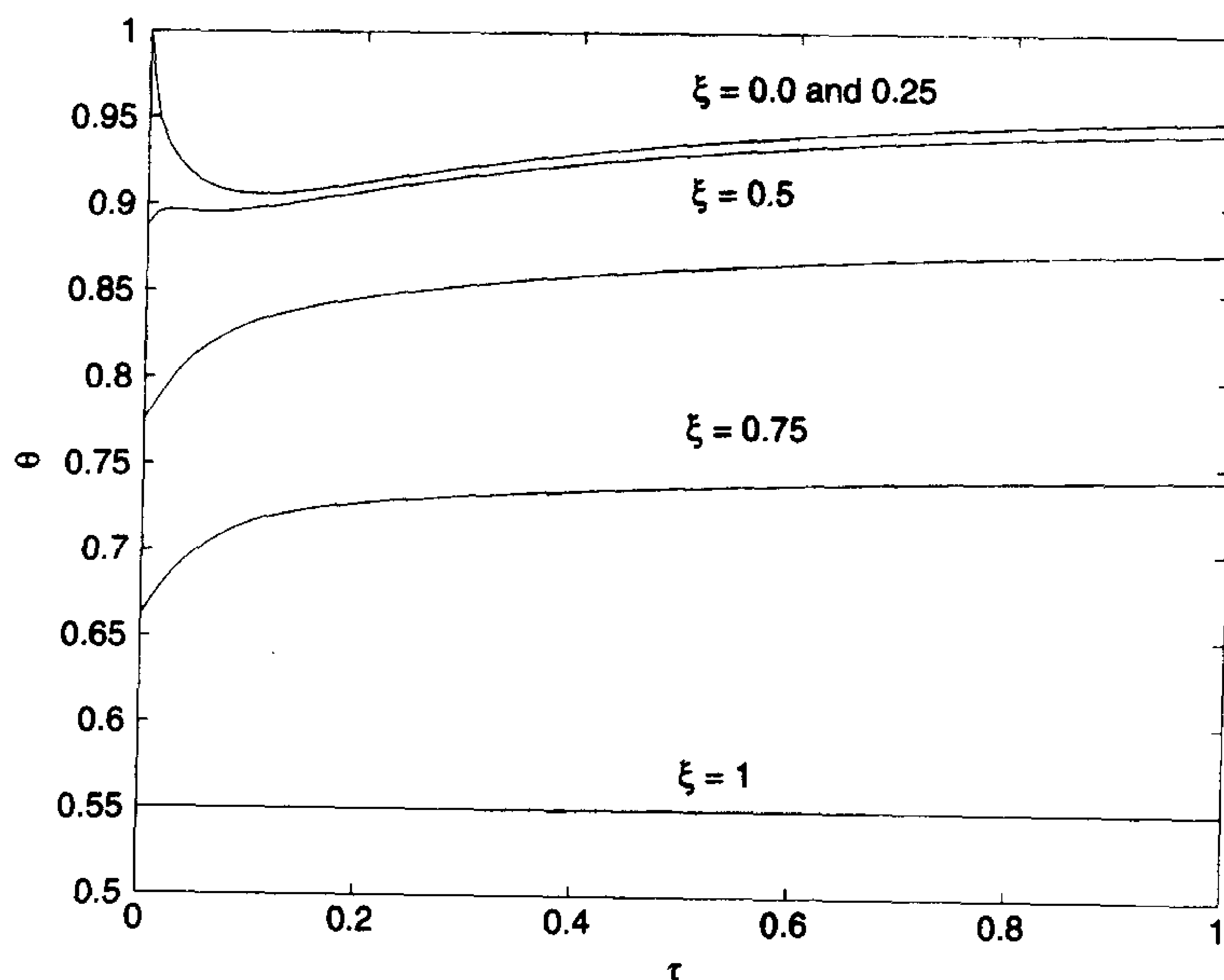


图 12.5　根据表 12.1 数据绘出的一维导热曲线图

12.2 壳管式热交换器的外形设计

根据表 12.2 和图 12.6 的定义,可以得出壳管式热交换器的特征方程。

表 12.2 与热交换器相关的符号定义

符号	单位	说明
c_p	J/(kg K)	恒压下的比热
d_o	m	管道外径
d_i	m	管道内径
f		管内流动摩擦系数
f_s		壳内流动摩擦系数
h_o	W/(m²K)	管内传热系数
h_i	W/(m²K)	管外传热系数
k	W/(m K)	流体热导函数
k_{tube}	W/(m K)	管道热导函数
$\dot{m}_t$	kg/s	管内质量流速
$\dot{m}_s$	kg/s	壳内质量流速
Δp_s	Pa	壳内压降
Δp_t	Pa	管内压降
u_t	M/s	管内流体的平均轴向速度($=\dot{m}_t/\rho A_t$)
A_o	m²	($=\pi d_o^4 LN_T/N_P$)管外表面积单位流量
A_i	m²	($=\pi d_i LN_T/N_P$)管内表面积单位流量
A_s	m²	壳中心线或靠近中心线的流体横截面积
A_t	m²	全部横截面的单位流量($=\pi d_i^2 N_T/(4N_P)$)
B	m	障板间距
C	m	相邻两管的距离(参见图 12.7)
C_L		管道的分布常数
C_{TP}		管道计算积累常数
D_e	m	壳等效直径
D_s	m	壳内径
F		逆流布置的对数平均温差(LMTD, log mean temperature difference)修正系数
L	m	管长
N_b		隔板数[(L/B)取整]
N_T		管数
N_P		管口数
P_T	m	节距(参见图 12.7)
P_P	W	管内流体的泵送功率
Pr		普朗特数
Q	W	传热率
R_{fo}	(m²K)/W	管外的污垢热阻
R_{fi}	(m²K)/W	管内的污垢热阻
Re_b		T_b 点的管内雷诺数
Re_s		T_b 点的壳内雷诺数
ΔT_m	℃,K	LMTD
T_{h_1}	℃,K	热流体的入口温度
T_{h_2}	℃,K	热流体的出口温度
T_{c_1}	℃,K	冷流体的入口温度
T_{c_2}	℃,K	冷流体的出口温度
T_b	℃,K	整体温度
T_w	℃,K	壁温
U	W/(m²K)	A 口的整体平均传热系数
η_P		泵效率($0.80\leq \eta_p \leq 0.85$)
ϕ_s		黏度修正系数
μ	kg/(s m)	动力学黏度
μ_b	kg/(s m)	T_b 点的动力学黏度
μ_w	kg/(s m)	T_w 点的动力学黏度
ρ	kg/m³	密度

热交换器的能量守恒方程为：

$$Q = UAF\Delta T_m = (\dot{m}c_p)_h(T_{h1} - T_{h2}) = (\dot{m}c_p)_c(T_{c2} - T_{c1})$$

其中，下标 1 代表流入，下标 2 代表流出，下标 h 代表热流体，下标 c 代表冷流体，假设逆流布置，对数平均温差如下：

$$\Delta T_m = \frac{\Delta T_1 - \Delta T_2}{\ln(\Delta T_1/\Delta T_2)}$$

$$\Delta T_1 = T_{h1} - T_{c2}$$

$$\Delta T_2 = T_{h2} - T_{c1}$$

图 12.6 典型单壳程－双管程的壳管式热交换器

图 12.7 两种管道铺设：(a)铺设角度 = 90°；(b)铺设角度 = 30°

对于只有一个壳程，2 个、4 个等管程的壳管式热交换器[①]，有下式：

$$F = R_s C_o \left[\log_{10} \frac{C_1 + R_s}{C_1 - R_s} \right]^{-1}$$

对于有两个壳程，4 个、8 个等管程的壳管式热交换器，有下式：

---

① 参见 R. A. Bowman, A. C. Mueller 和 W. M. Nagle, uMean Temperature Difference in Design," *Transaction of ASME*, 第 62 卷, 1940 年 5 月, 第 283～293 页。

$$F = 0.5R_s C_o\left[\log_{10}\frac{C_1 + C_2 + R_s}{C_1 + C_2 - R_s}\right]^{-1}$$

其中，

$$R_s = \sqrt{R^2 + 1} \quad P = \frac{T_{c2} - T_{c1}}{T_{h1} - T_{c1}} \quad R = \frac{T_{h1} - T_{h2}}{T_{c2} - T_{c1}}$$

$$C_1 = \frac{2}{P} - 1 - R \quad C_2 = \frac{2}{P}\sqrt{(1 - P)(1 - PR)}$$

$$C_o = \frac{1}{R - 1}\lg\frac{1 - P}{1 - PR} \quad R \neq 1$$

$$= \frac{P}{2.3(1 - P)} \quad R = 1$$

总传热系数 U 由下式给出：

$$U = \left[\frac{d_o}{d_i}\frac{1}{h_i} + \frac{d_o}{d_i}R_{fi} + \frac{d_o}{2k_{tube}}\ln\frac{d_o}{d_i} + R_{fo} + \frac{1}{h_o}\right]^{-1}$$

管内的传热系数值近似为：

$$h_i = \frac{k}{d_i}Nu_b$$

这里由于通过环形管道流体的性质不变，对于强制对流的充分发展的紊流流动有下式①：

$$Nu_b = \frac{0.125f(\mathrm{Re}_b - 1000)\mathrm{Pr}_b}{1 + 12.7\sqrt{0.125f((\mathrm{Pr}_b)^{2/3} - 1)}} \quad 0.5 < \mathrm{Pr}_b < 2000 \quad 3000 < \mathrm{Re}_b < 5 \times 10^6$$

下标 b 表明该值是在管道内流体平均温度基础上计算出来的，称为流体平均温度。

f 是在光滑管道内的流动摩擦系数：

$$f = (0.790\ \ln\mathrm{Re}_b - 1.64)^{-2} \quad 3000 \leqslant \mathrm{Re}_b \leqslant 10^6$$

对于不光滑管道，令 $\lambda = f$，利用习题 5.19 的结果。

雷诺数 Re_b 由下式给出：

$$\mathrm{Re}_b = \frac{\rho u_t d_i}{\mu} = \frac{4\dot{m}_t}{\pi d_i \mu N_T / N_p}$$

式中，

$$u_t = \dot{m}_t / (0.25\rho\pi d_i^2 N_T / N_p)$$

普朗特数为 Pr_b，由下式给出：

$$\mathrm{Pr}_b = \frac{c_p\mu}{k}$$

壳内的传热系数可以用下式估计：

$$h_o = 0.36\frac{k\phi_s}{D_e}(\mathrm{Re}_s)^{0.55}(\mathrm{Pr})^{1/3} \qquad 2000 < \mathrm{Re}_s < 10^6$$

$$\phi_s = \left(\frac{\mu_b}{\mu_w}\right)^{0.14} \qquad T_w = (T_{c1} + T_{c2} + T_{h1} + T_{h2})/4$$

除了 ϕ_s 外，在壳内流体平均温度基础上算出其他值，在 T_w 温度时算出 μ_w。管道在壳内成正

① 参见 F.P.Incropera 和 D.P.DeWitt，出处同上，第 424 页。

方形铺设,如图 12.7 所示,壳的等效直径是 D_e,其表达式如下:

$$D_e = 4(P_T^2 - \pi d_0^2/4)/\pi d_o$$

对于管道成三角形铺设的情况:

$$D_e = 8(P_T^2\sqrt{3}/4 - \pi d_0^2/8)/\pi d_o$$

壳内雷诺数 Re_s 由下式给出:

$$\mathrm{Re}_s = \frac{\dot{m}_s D_e}{A_s \mu}$$

$$A_s = 1.128CB\sqrt{\frac{N_T C_L}{C_{TP}}} = \frac{D_s CB}{P_T}$$

式中,D_s 是壳内径:

$$D_s = 1.128P_T\sqrt{\frac{N_T C_L}{C_{TP}}}$$

设 C_{TP}的值分别为:

管程数为 1($N_P = 1$):$C_{TP} = 0.93$

管程数为 2($N_P = 2$):$C_{TP} = 0.90$

管程数为 3($N_P = 3$):$C_{TP} = 0.85$

C_L 的值分别为(参见图 12.7):

$C_L = 1$ 对应于 90°和 45°

$C_L = 0.87$ 对应于 30°和 60°

给定壳内径(D_s)作为管道外径(d_o)的函数,标准管道铺设表可以给出所需管道数(N_T)、管道的节距(正方形或三角形)和管道数(N_P)①。因此,数值计算结果通常接近标准值。

壳内的压降由下式给出:

$$\Delta p_s = \frac{f_s \dot{m}_s^2 (N_b + 1) D_s}{2A_s^2 \rho D_e \phi_s}$$

其中,

$$f_s = \exp(0.576 - 0.19 \ln \mathrm{Re}_s)$$

在温度为 T_b 时计算出这些值,管内的压降为:

$$\Delta p_t = f\left(\frac{LN_p}{d_i}\right)\left(\frac{\rho u_t^2}{2}\right)$$

忽略管道的弯曲因素,在温度 T_b 时计算出式中的值,仅适用于低速流动液体。

泵送功率是:

$$P_P = \frac{\dot{m}_t \Delta p_t}{\rho \eta_p}$$

举例说明这些结果的应用。

---

① 参见 S. Kakac 和 H. Liu, *Heat Exchangers: Selection, Rating, and Thermal Design*, CRC Press, Boca Raton, FL, 1998 年,第 258~261 页。

例 12.4 确定管长和压降

用壳程为 1,管程为 2 的壳管式热交换器进行水水热交换,放热大约 800 kW。管道内是冷水,壳内是热水,初始分析结果参见表 12.3。分析目标是:热交换器的长度是否可以小于 4.5 m,壳内和管内的压降是否可以小于 4000 Pa。重新整理上面给出的方程,得出下式:

$$T_{h_2} = T_{h_1} - \frac{(\dot{m}c_p)_t}{(\dot{m}c_p)_s}(T_{c_2} - T_{c_1})$$

$$\Delta p_t = C_o \frac{f\dot{m}_t^3}{UF\Delta T_m}$$

$$\Delta p_s = \frac{f_s\dot{m}_s^2(N_b + 1)D_s}{2A_s^2\rho D_e\phi_s}$$

$$L = \frac{(\dot{m}c_p)_t(T_{c_2} - T_{c_1})}{\pi d_o N_T UF\Delta T_m}$$

式中,

$$C_o = \frac{N_P(T_{c_2} - T_{c_1})(c_p)_t}{2\pi d_i d_o N_T \rho_t A_t^2}$$

$$A_t = \frac{\pi d_i^2}{4}\frac{N_T}{N_P}$$

表 12.3 热交换器初步分析的参数

几何参数	物理参数
$D_s = 0.39$ m	$T_{c_1} = 18$℃
$N_T = 124$	$T_{c_2} = 42$℃
$N_P = 2$	
$P_T = 0.024$ m	$T_{h_1} = 65$℃
$B = 0.5$ m	
$N_b = 4$	$\dot{m}_s = \dot{m}_h = 14$ kg/s
$d_i = 16$ mm(0.016 m)	$\dot{m}_t = \dot{m}_c = 8.5$ kg/s
$d_o = 19$ mm(0.019 m)	$R_{fi} = 0.000\ 15$
$k_{tube} = 60$ W/m² K(碳钢)	$R_{fo} = 0.000\ 15$
90°管道铺设	
管道辅设成正方形	
光滑管道	

表 12.4 例 12.1 中的正函数

函数名	用途
LMTDcorrFacto	计算 F
TubeFF	计算光滑管道的 f
WaterProperties	计算从 0 到 100℃间任意温度点上 k, ρ, μ, c_p 和 Pr 的值
LMTD	计算 LMTD
hTubeOutside	计算 h_o 和 Δp_s
hTubeInside	计算 h_i
PressureDropLength	计算 Δp_t 和 L
T2HotSide	使用 fzero 计算 T_{h_2}

表 12.5 水的热物理性能

$T(K)$	k(W/m K)	ρ(kg/m³)	μ(N s/m²)	c_p(J/kg K)	Pr
273	0.569	1000.0	1750×10^{-6}	4217	12.99
285	0.590	1000.0	1225×10^{-6}	4189	8.81
300	0.613	998.0	855×10^{-6}	4179	5.83
315	0.634	991.1	631×10^{-6}	4179	4.16
330	0.650	984.3	489×10^{-6}	4184	3.15
345	0.668	976.6	389×10^{-6}	4191	2.45
360	0.674	967.1	324×10^{-6}	4203	2.02
373	0.680	957.9	279×10^{-6}	4217	1.76

用 **HeatExchanger** 函数来求解，其中包括 8 个子函数，如表 12.4 所示。用表 12.5 给出的数据确定水的各种物理性能。

```
function HeatExchanger
global Tc1 Tc2 Th1 ms mt
global DS C B PT pitch
global NP NT
global di do kTube Rfi Rfo Nb
Ds = 0.39;NT = 124;NP = 2;B = 0.5;PT = 0.024;
di = 0.016;do = 0.019;kTube = 60;
Rfi = 0.00015;Rfo = 0.00015;
pitch = 'square';C = PT-do;ms = 14;mt = 8.5;
Tc1 = 18;Tc2 = 42;Th1 = 65;Nb = 4;
options = optimset('Display','off');
Th2 = fzero(@T2HotSide,Th1 + 15,options);
[DeltaPt,DeltaPs,L] = PressureDropLength(Th2);
disp(['Shell-side exit temperature = 'num2str(Th2)'deg C'])
disp(['Tube-side pressure drop = 'num2str(DeltaPt)'Pa'])
disp(['Shell-side pressure drop = 'num2str(DeltaPa)'Pa'])
disp(['Tube length = 'num2str(L)'m'])

function F = LMTDcorrFactor(Th2)
global Tc1 Tc2 Th1 ms mt
global NP NT
P = (Tc2 - Tc1)/(Th1 - Tc1);
R = (Th1 - Th2)/(Tc2 - Tc1);
C1 = 2/P - 1 - R;
C2 = 2 * sqrt((1 - P) * (1 - P * R))/P;
Rs = sqrt(R^2 + 1);
if R == 1
  Co = R/2.3/(1 - P);
else
  Co = log10((1 - P)/(1 - P * R))/(R - 1);
```

```
end
if NP == 1
  F = Rs * Co/log10((C1 + Rs)/(C1 - Rs));
else
  F = 0.5 * Rs * Co/log10((C1 + C2 + Rs)/(C1 + C2 - Rs));
end

function f = TubeFF(Re)
f = 1/(0.79 * log(Re) - 1.64)^2;

function[cp,mu,k,rho,Pr] = WaterProperties(Temp)
Temp = Temp + 273;
T = [273,285,300,315,330,345,360,373];
cpp = [4217,4189,4179,4179,4184,4191,4203,4217];
muu = [1750,1225,855,631,489,389,324,279] * 1e-6;
kk = [569,590,613,634,650,668,674,680] * 0.001;
rhoo = [1000,1000,998,991.1,984.3,976.6,967.1,957.9];
Prr = [12.99,8.81,5.83,4.16,3.15,2.45,2.02,1.76];
cp = spline(T,cpp,Temp);
mu = spline(T,muu,Temp);
k = spline(T,kk,Temp);
rho = spline(T,rhoo,Temp);
Pr = spline(T,Prr,Temp);

function Tm = LMTD(Th2)
global Tc1 Tc2 Th1 ms mt
DT1 = Th1 - Tc2;
DT2 = Th2 - Tc1;
Tm = (DT1 - DT2)/log(DT1/DT2);

function hi = hTubeInside(Reb,Prb,kb)
global di do kTube Rfi Rfo Nb
f = TubeFF(Reb);
Nub = 0.125 * f * (Reb - 1000) * Prb/(1 + 12.7 * sqrt(0.125 * f) * (Prb^(2/3) - 1));
hi = Nub * kb/di;

function[ho,DeltaPs] = hTubeOutside(Tb,Tw)
global Ds C B PT pitch
global di do kTube Rfi Rfo Nb
global Tc1 Tc2 Th1 ms mt
[cpb,mub,kb,rhob,Prb] = WaterProperties(Tb);
[cpw,muw,kw,rhow,Prw] = WaterProperties(Tw);
phis = (mub/muw)^0.14;
if pitch == 'square'
  De = 4 * (PT ^ 2-pi * do^2/4)/pi/do;
else
  De = 8 * (PT ^ (sqrt(3)/4)-pi * do^2/8)/pi/do;
end
As = Ds * C * B/PT;
Res = ms * De/mub/As;
ho = 0.36 * kb * phis * Res^0.55 * Prb^(1/3)/De;
fs = exp(0.576-0.19 * log(Res));
DeltaPs = fs * ms^2 * (Nb + 1) * Ds/(2 * As^2 * rhob * De * phis);

function Th = T2HotSide(Th2)
```

```
global Tc1 Tc2 Th1 ms mt
cph = WaterProperties((Th2 + Th1)/2);
cpc = WaterProperties((Tc1 + Tc2)/2);
Th = Th1 - mt * cpc * (Tc2 - Tc1)/ms/cph - Th2;

function [DeltaPt, DeltaPs, L] = PressureDropLength(Th2)
global Tc1 Tc2 Th1 ms mt
global DS C B PT pitch
global NP NT
global di do kTube Rfi Rfo Nb
Tcb = (Tc1 + Tc2)/2;
[cpc, muc, kc, rhoc, Prc] = WaterProperties(Tcb);
[cph, muh, kh, rhoh, Prh] = WaterProperties((Th1 + Th2)/2);
Th2 = Th1 - mt * cpc * (Tc2 - Tc1)/ms/cph;
Thb = (Th1 + Th2)/2;
At = 0.25 * pi * di^2 * NT/NP;
Co = 2 * NP * (Tc2 - Tc1) * cpc/pi/di/do/NT/rhoc/At^2;
Tm = LMTD(Th2);
F = LMTDcorrFactor(Th2);
Tw = (Th1 + Th2 + Tc1 + Tc2)/4;
[ho, DeltaPs] = hTubeOutside(Thb, Tw);
Rec = 4 * mt * NP/pi/di/muc/NT;
hi = hTubeInsIde(Rec, Prc, kc);
U = 1/(do/di/hi + do * Rfi/di + do/2/kTube * log(do/di) + Rfo + 1/ho);
f = TubeFF(Rec);
DeltaPt = Co * f * mt^3/U/F/Tm;
Q = mt * cpc * (Tc2 - Tc1);
L = Q/pi/do/NT/U/F/Tm;
```

程序执行后的结果在 Matlab 命令窗口中显示出来：

```
Shell-side exit temperature = 50.45 deg C
Tube-side pressure drop = 14274.54 Pa
Shell-side pressure drop = 1799.40 Pa
Tube length = 4.232 m
```

可以看出该结果满足设计要求。

12.3 对流传热

12.3.1 平板热边界层——相似解

图 12.8 给出了流过平板的层流边界层的速度纵剖图，该图通过求解下面的勃拉休斯方程获得[①]：

$$\frac{d^3 f}{d\eta^3} + \frac{f}{2}\frac{d^2 f}{d\eta^2} = 0$$

① 参见 F.P.Incropera 和 D.P.DeWitt，出处同上，第 350～352 页。

其中,f是一个修正的流函数:

$$f = \frac{\Psi}{u_\infty \sqrt{v_{vis} x / u_\infty}}$$

流函数 Ψ 定义为:

$$u = \partial \Psi / \partial y$$
$$v = - \partial \Psi / \partial x$$

其中,u 和 v 分别是对应 x 和 y 方向上的速度,η 是相似变量:

$$\eta = y \sqrt{u_\infty / v_{vis} x}$$

来流速度为 u_∞,流体的运动黏度是 v_{vis}。通过求解勃拉休斯方程可以得到边界层上任意一点的速度。11.3.2 节给出了勃拉休斯方程的数值解。

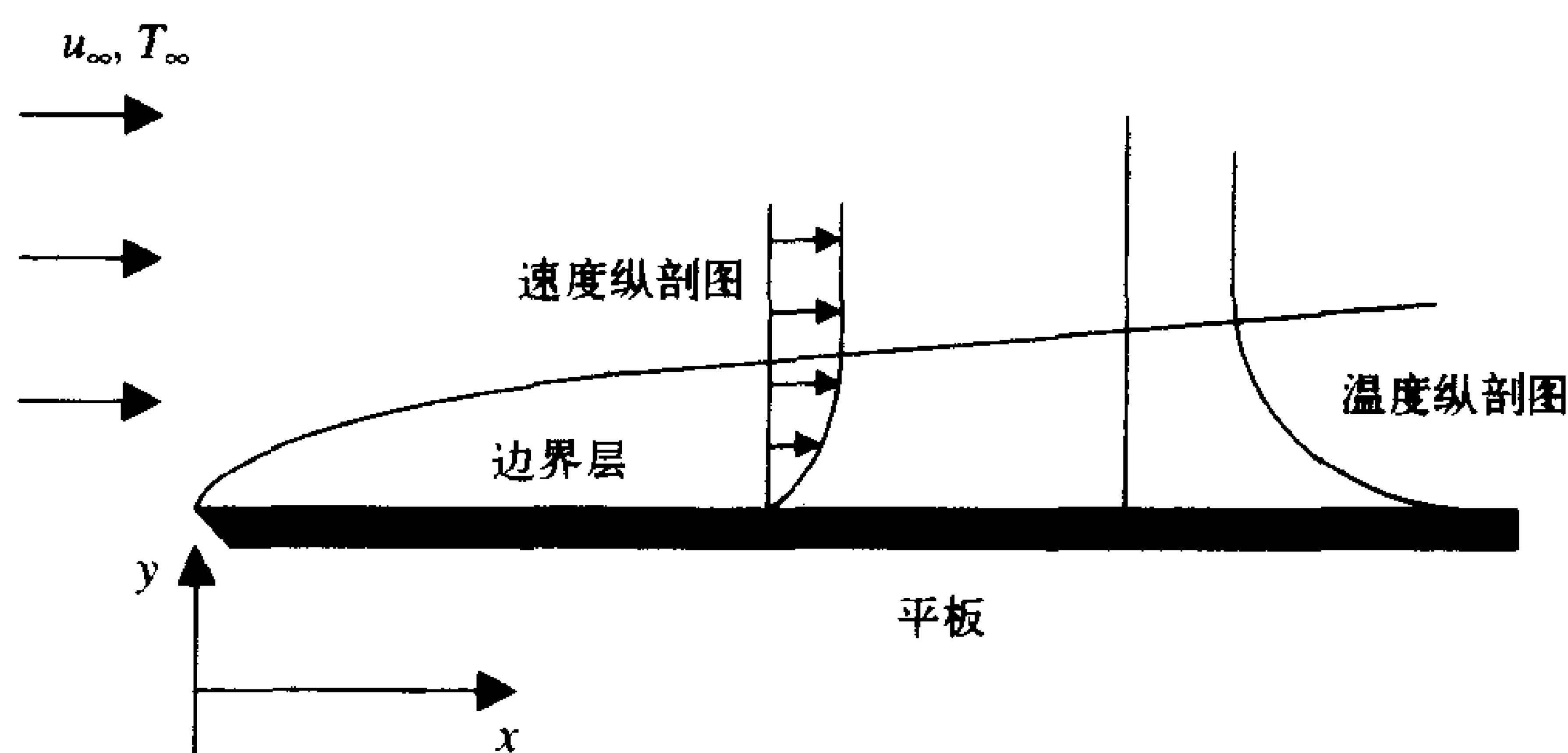


图 12.8 平板上的流动

在稳流的情况下,对边界层进行一些假设,则流体的热能方程可以通过相似变量获得[①]:

$$\frac{d^2 T^*}{d\eta^2} + \Pr \frac{f}{2} \frac{dT^*}{d\eta} = 0$$

其中,T^* 是无量纲的温度值:

$$T^* = \frac{T - T_s}{T_\infty - T_s}$$

T 是流体温度,T_s 是平板表面温度,T_∞ 是流体的来流温度,普朗特数是 $\Pr = v_{vis}/\alpha$,其中 α 是流体的热扩散率。注意,通过能量方程中的 f, T^* 和流速耦合。

边界条件为:

$$f(0) = 0 \quad \left.\frac{df}{d\eta}\right|_{\eta=0} = 0 \quad \left.\frac{df}{d\eta}\right|_{\eta=\infty} \to 1$$
$$T^*(0) = 0 \quad T^*(\eta \to \infty) \to 1$$

例 12.5 平板上的流动:勃拉休斯方程

当 $\Pr = 0.07, 0.7$ 和 7.0 时求平板上流动的勃拉休斯方程的解。利用 bvp4c 函数,通过假定 η 的最大值(η_{max})使当 $\eta \to \infty$ 时边界条件近似满足,可以求得解。通过引入以下变量可以将这两个相关的非线性方程转换为五个相关的一阶常微分方程:

$$y_1 = f \qquad\qquad y_4 = T^*$$

① 参见 F.P.Incropera 和 D.P.DeWitt,出处同上。

$$y_2 = \frac{df}{d\eta} \qquad y_5 = \frac{dT^*}{d\eta}$$

$$y_3 = \frac{d^2f}{d\eta^2}$$

其中，y_1 为流函数，y_2 为速度，y_3 是变量，y_4 为温度，y_5 为热流量。

这些量由五个一阶微分方程决定：

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_4}{d\eta} = y_5$$

$$\frac{dy_2}{d\eta} = y_3 \qquad \frac{dy_5}{d\eta} = -\frac{Pr}{2}y_1y_5$$

$$\frac{dy_3}{d\eta} = -\frac{1}{2}y_1y_3$$

相应的边界条件为：

$$y_1(0) = 0 \qquad y_4(\eta \to \infty) \to 1$$

$$y_2(0) = 0 \qquad y_4(0) = 0$$

$$y_2(\eta \to \infty) \to 1$$

创建主函数 **BlasiusX**，其中包括支持 bvp4c 的两个子函数。函数 **BlasiusT** 用来求解五个一阶微分方程，函数 **BlasiusTbc** 用来求出边界条件。当近似有 $\eta \to \infty$ 时，用 $\eta_{max} = 8$ 对应 Pr = 0.7 和 7.0；$\eta_{max} = 20$ 对应 Pr = 0.07。

```
function BlasiusX
Pr=[0.07,0.7,7.0];etaMax=[1,5,8,8];xm=[1,5,5,5];
for k=1:3
 figure(k)
 solinit=bvpinit(linspace(0,etaMax(k),8),[0,0,0,0,0]);
 sol=bvp4c(@BlasiusT,@BlasiusTbc,solinit,[],Pr(k));
 eta=linspace(0,etaMax(k));
 y=deval(sol,eta);
 subplot(2,1,1)
 plot(tea,y(1,:),'-.k',eta,y(2,:),'-k',eta,y(3,:),'--k')
 xlabel('\eta')
 ylabel('y_1(i=1,2,3)')
 legend('Stream function,f=y_1','Velocity,df/d\eta=y_2',
  'Shear,d_2f/d\eta_2=y_3')
 axis([0 xm(k)0 2])
 subplot(2,1,2)
 plot(eta,y(4,:),'-k',eta,y(5,:),'--k')
 axis([0 xm(k)0 2])
 legend('Temperature,T_^*=y_4','Heat flux,dT_^*/d\eta=y_5')
 xlabel('\eta')
 ylabel('y_i(i=4,5)')
end
function F=BlasiusT(eta,y,Pr)
F=[y(2);y(3);-0.5*y(1)*y(3);y(5);-Pr*0.5*y(1)*y(5)];
function res=BlasiusTbc(ya,yb,Pr)
res=[ya(1);ya(2);ya(4);yb(2)-1;yb(4)-1];
```

程序执行后的结果参见图 12.9 至图 12.11。

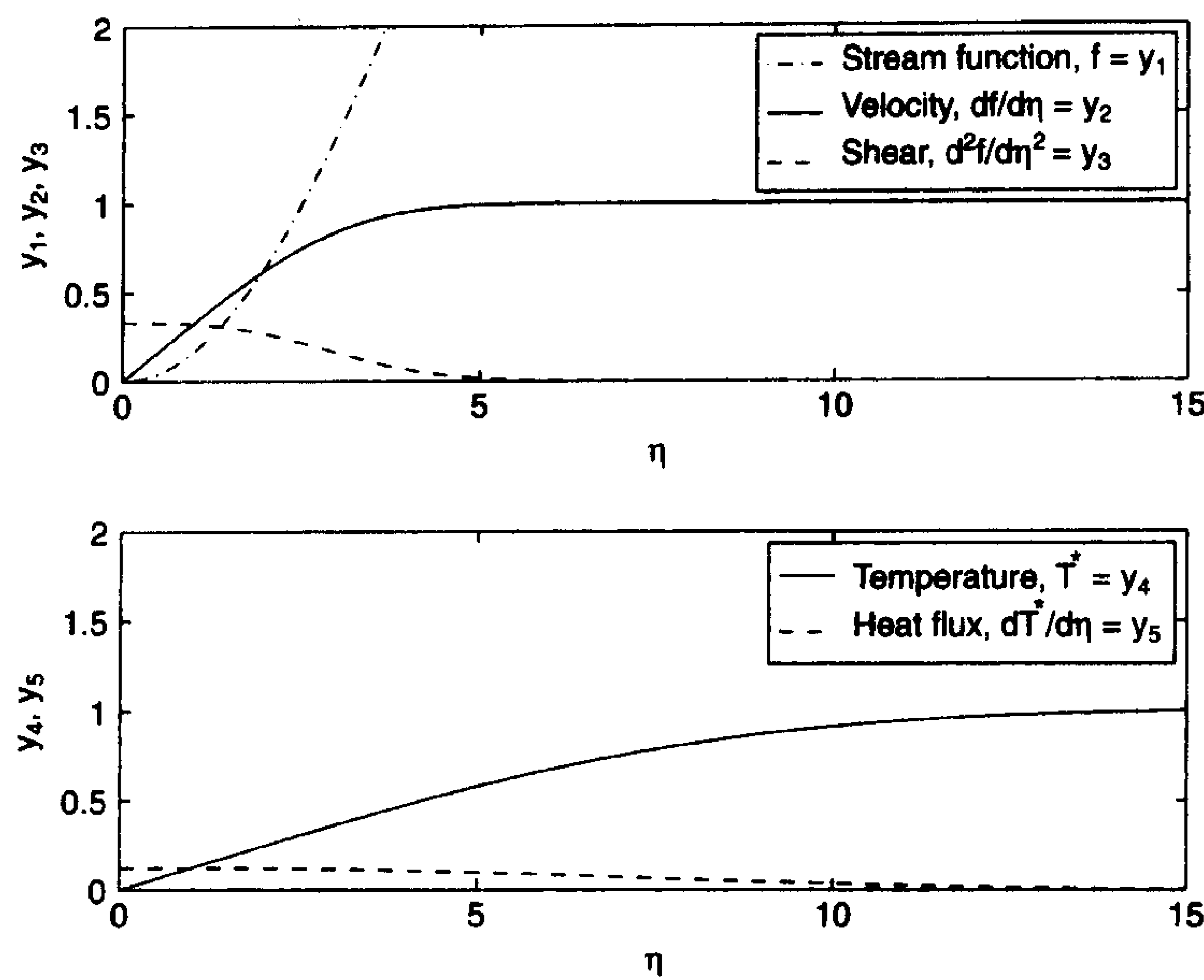


图 12.9 Pr = 0.07 时扩展勃拉休斯方程的解

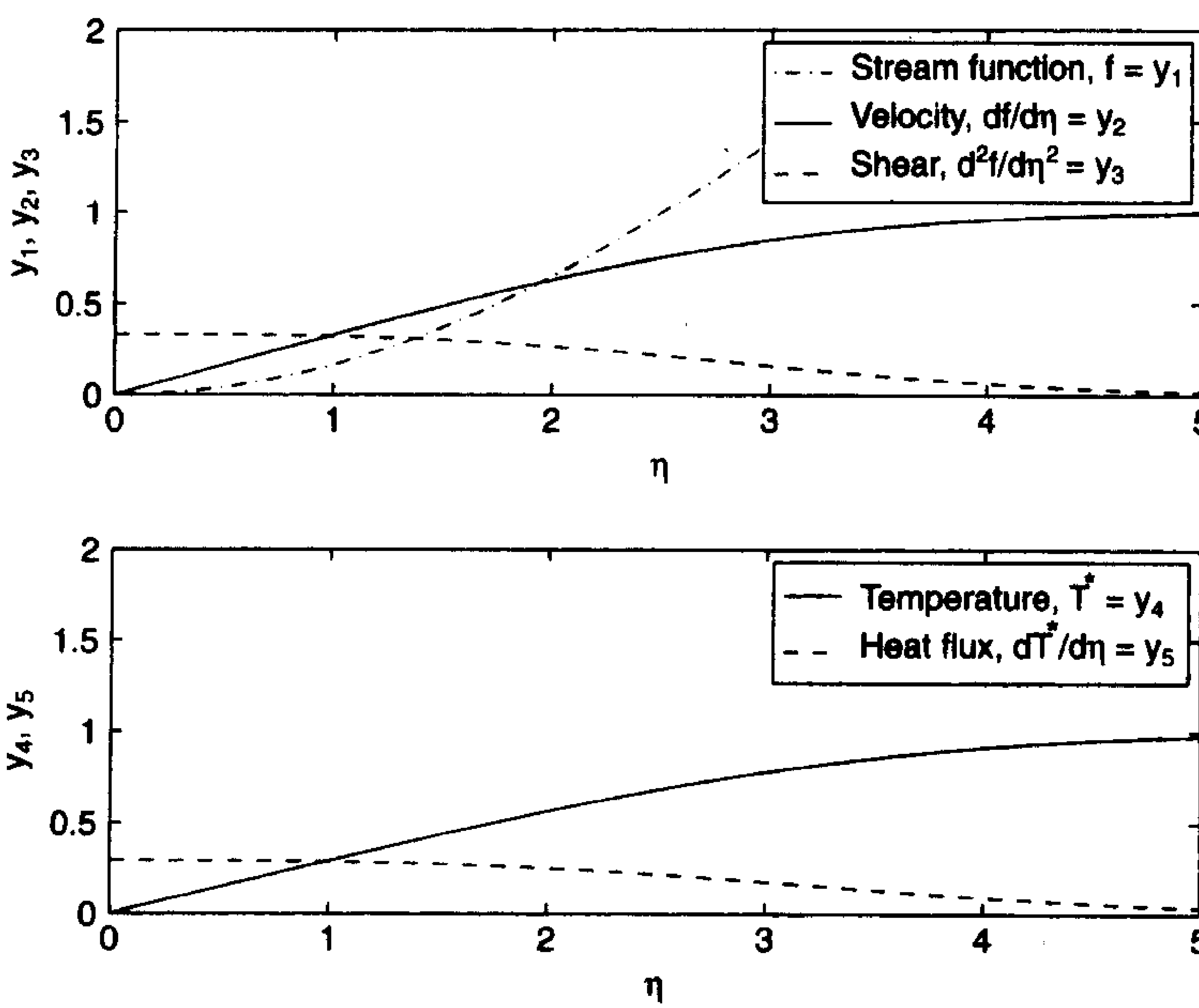


图 12.10 Pr = 0.7 时扩展勃拉休斯方程的解

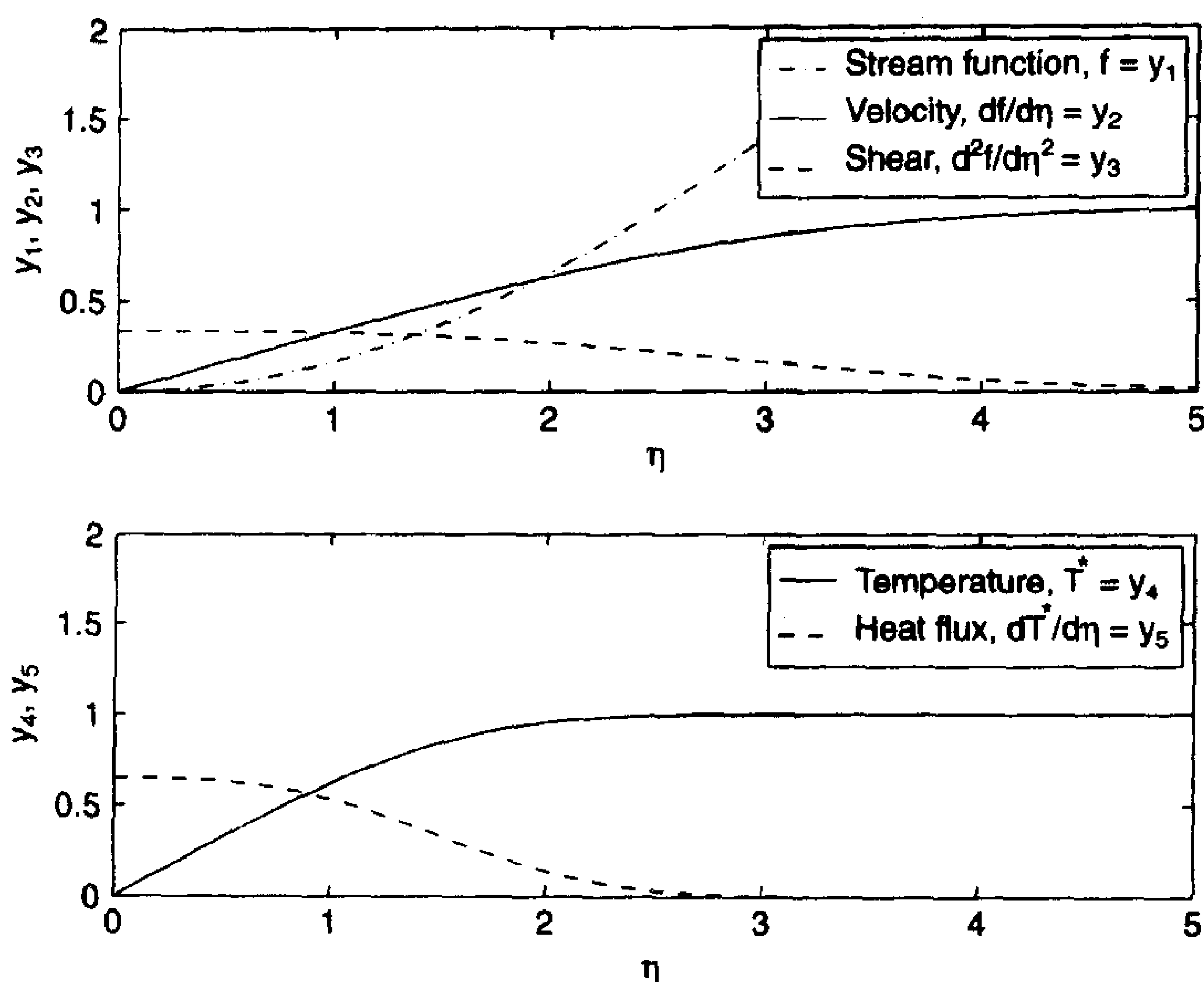


图 12.11　Pr = 7.0 时扩展勃拉休斯方程的解

12.3.2　自然对流相似解

竖直的加热板与冷的液体相接触，它们之间的自然对流如图 12.12 所示。大部分的流体是静止的，但是由于平板上的热交换造成浮升力流动。这种流动可以由下面给出的两个相关的非线性常微分方程描述[①]：

$$\frac{d^3 f}{d\eta^3} + 3f\frac{d^2 f}{d\eta^2} - 2\left(\frac{df}{d\eta}\right)^2 + T^* = 0$$

$$\frac{d^{2*} T^*}{d\eta^2} + 3\mathrm{Pr} f\frac{dT^*}{d\eta} = 0$$

式中，f 是修正流函数：

$$f = \frac{\Psi}{4v_{vis}(Gr_x/4)^{0.25}}$$

流函数 Ψ 定义为：

$$u = \partial\Psi/\partial y$$

$$v = -\partial\Psi/\partial x$$

式中，u 和 v 分别为沿 x 和 y 方向上的速度。η 为相似变量：

$$\eta = \frac{y}{x}\left(\frac{Gr_x}{4}\right)^{0.25}$$

格拉晓夫(Grashof)数定义如下：

$$Gr_x = g\beta(T_s - T_\infty)x^3/v_{vis}^2$$

式中，g 是重力加速度，β 是热膨胀系数，为：

$$\beta = \frac{-1}{\rho}\left(\frac{\partial\rho}{\partial T}\right)_p$$

① 参见 F.P.Incropera 和 D.P.DeWitt，出处同上，第 487 ~ 490 页。

v_{vis}是运动黏度，Pr 是先前定义的普朗特数，T^* 是归一化温度，表达式如下：

$$T^* = \frac{T - T_\infty}{T_s - T_\infty}$$

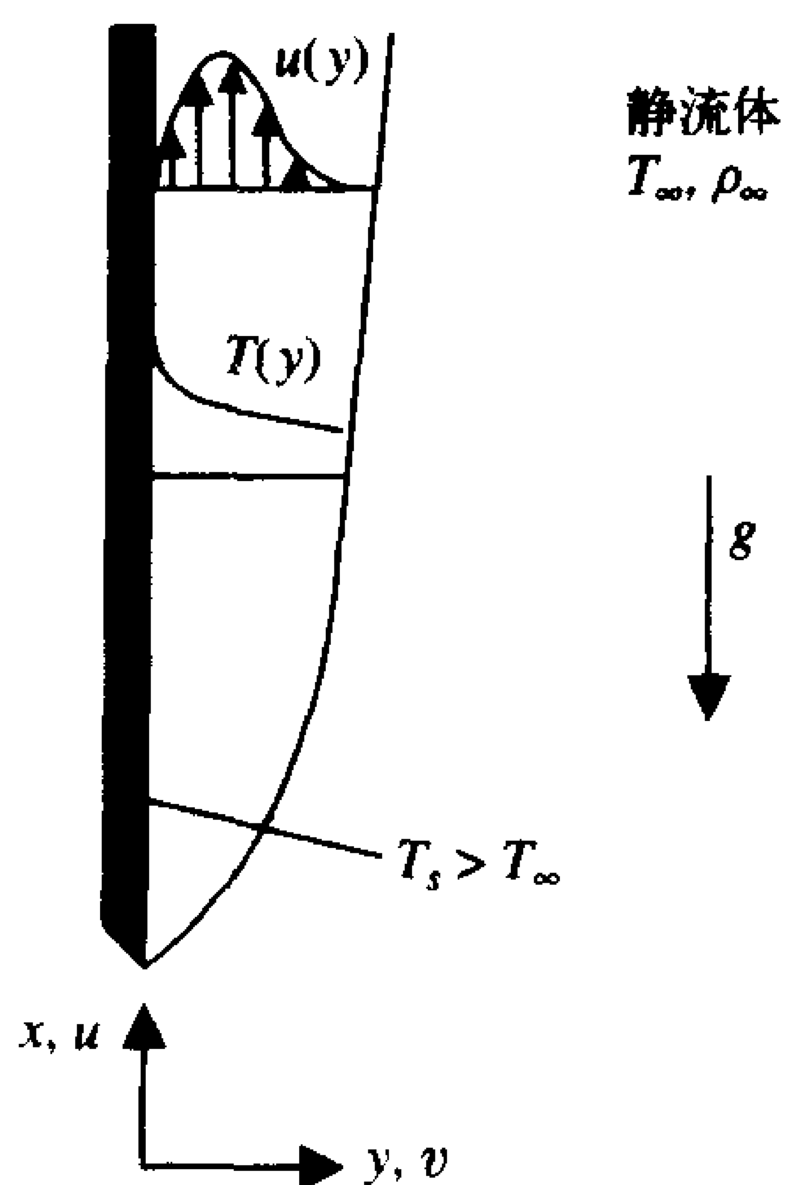


图 12.12 加热平板的自然对流

该系统的边界条件为：

$\eta = 0$:

$$f = 0, \frac{df}{d\eta} = 0, T^* = 1$$

$\eta \to \infty$:

$$\frac{df}{d\eta} \to 0 \quad T^* \to 0$$

通过引入下面的中间变量，可以将前面的系统用五个一阶微分方程表示：

$$y_1 = f \qquad y_4 = T^*$$

$$y_2 = \frac{df}{d\eta} \qquad y_5 = \frac{dT^*}{d\eta}$$

$$y_3 = \frac{d^2 f}{d\eta^2}$$

含有新变量的微分方程如下：

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_4}{d\eta} = y_5$$

$$\frac{dy_2}{d\eta} = y_3 \qquad \frac{dy_5}{d\eta} = -3\mathrm{Pr}\, y_1 y_5$$

$$\frac{dy_3}{d\eta} = 2y_2^2 - 3y_1 y_3 - y_4$$

相应的边界条件为：

$$y_1(0) = 0 \qquad y_4(0) = 1$$

$$y_2(0) = 0 \qquad y_4(\eta \to \infty) \to 0$$

$$y_2(\eta \to \infty) \to 0$$

例 12.6 加热平板上的自然对流

当 Pr = 0.07,0.7 和 7.0 时,求加热平板上描述自然对流的方程组的解。$\eta \to \infty$ 时近似满足条件,此时的 η 值标记为 η_{max}。η_{max} 取决于普朗特数,当普朗特数较大时,η_{max} 相应减小。当结果与 η_{max} 的取值无关时,即可得到精确的解。在本例中,用 η_{max} = 8 分别对应 Pr = 0.7和 7; η_{max} = 11 对应 Pr = 0.07。

使用 bvp4c 来求解边界值问题。在这过程中需要创建两个函数:函数 **NaturalConv** 用于计算五个常微分方程中的各个系数;**NaturalConvbc** 用于求解边界条件。用于求解的主函数和子函数如下所示:

```
function NaturalConvection
Pr=[.07.7 7];eta_max=[11 8 8];xm=[10 5 5];ym=[2 0.8 0.5];
guess=[0 0 0 0 0];
for k=1:3
 figure(k);;
 solinit=bvpinit(linspace(0,eta_max(k),5),guess);
 sol=bvp4c(@NatConv,@NatConvBC,solinit,[],Pr(k));
 eta=linspace(0,eta_max(k),300);
 y=deval(sol,eta);
 subplot(2,1,1)
 plot(eta,y,c1,:),'-k',eta,y(2,:),eta,y(3,:),'--k')
 legend('Stream function,f=y_1','Veloeity,df/d\eta=y_2,',
  'Shear d_2f/d\eta_2=y_3,')
 axis([0 xm(k)-0.2 ym(k)])
 xlabel('\eta')
 ylabel('y_1,y_2,y_3)
 subplot(2,1,2)
 plot(eta,y(4,:),'-k',eta,y(5,:),'--k')
 legend('Temperature,T_^*=y_4','Heat flux,dT_^*/d\eta=y_5')
 axis([0 xm(k)-1.2 1])
 xlabel('\eta')
 ylabel('y_4,y_5')
end

function ff=NatConv(eta,y,Pr)
ff=[y(2);y(3);-3*y(1)*y(3)+2*y(2)^2-y(4);y(5);-3*Pr*y(1)*y(5)];

function res=NatConvBC(ya,yb,Pr)
res=[ya(1);ya(2);ya(4)-1;yb(2);yb(4)];
```

执行下面的程序得到如图 12.13 至图 12.15 所示的解。参见这三幅图,分别对应的普朗特数为 0.07, 0.7 和 7.0。可以看出,对以上三种情况,η 在 1 附近时速度达到最大值。流体中平行于壁的剪切应力为:

$$\tau_s = \frac{\sqrt{2}v^2\rho}{x^2} Gr_x^{3/4} \frac{d^2 f}{d\eta^2}$$

在速度达到最大值时上式趋于 0。

热导致流体流动,其结果是温度和速度边界层的厚度近似并与普朗特数无关。这一点与 12.3.1 节中流过平板的例子中的结果不同。该例中,当 Pr = 0.07 时速度边界层厚度远小于温度边界层厚度。

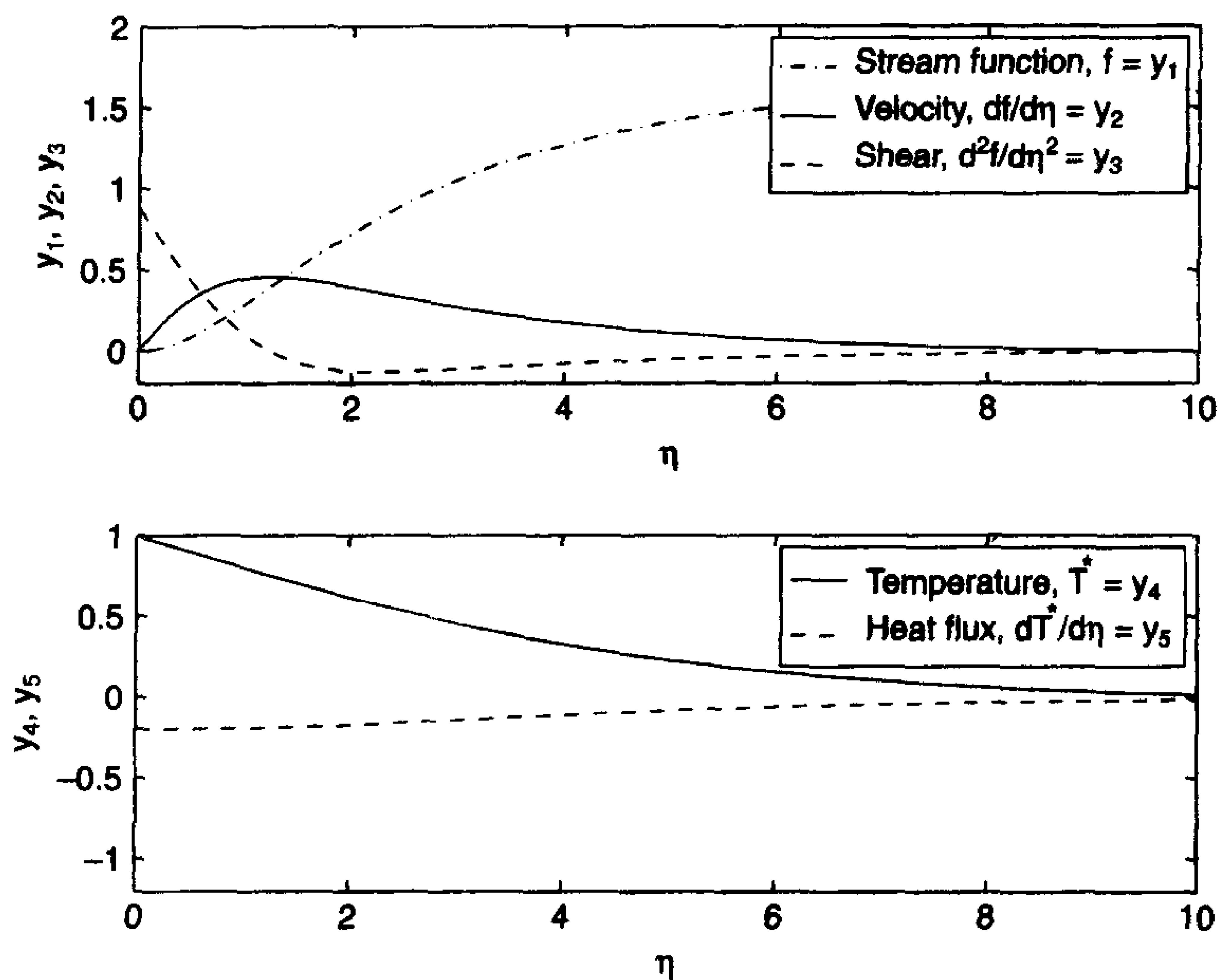


图 12.13 当 Pr=0.07 时的自然对流的解

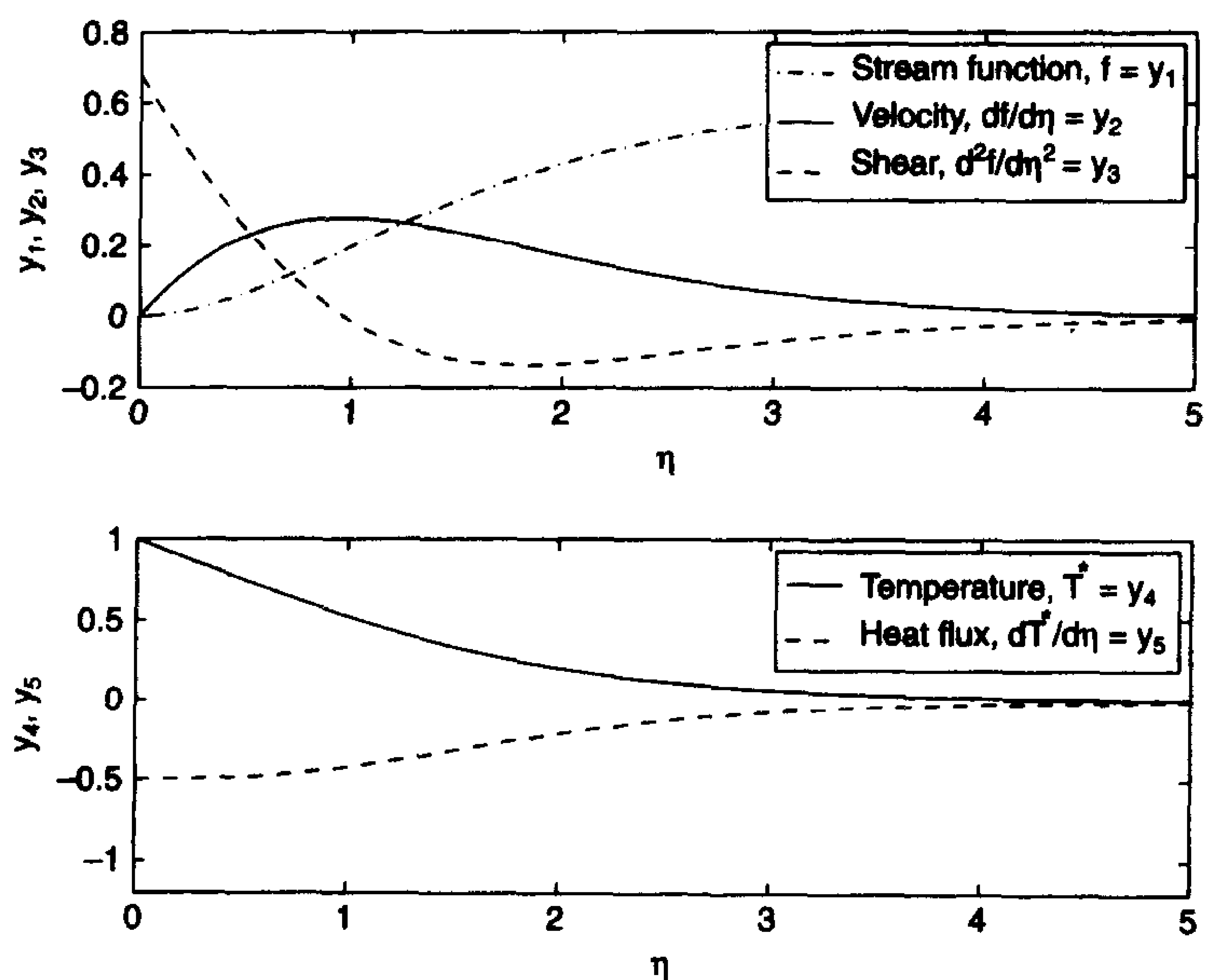


图 12.14 当 Pr=0.7 时的自然对流的解

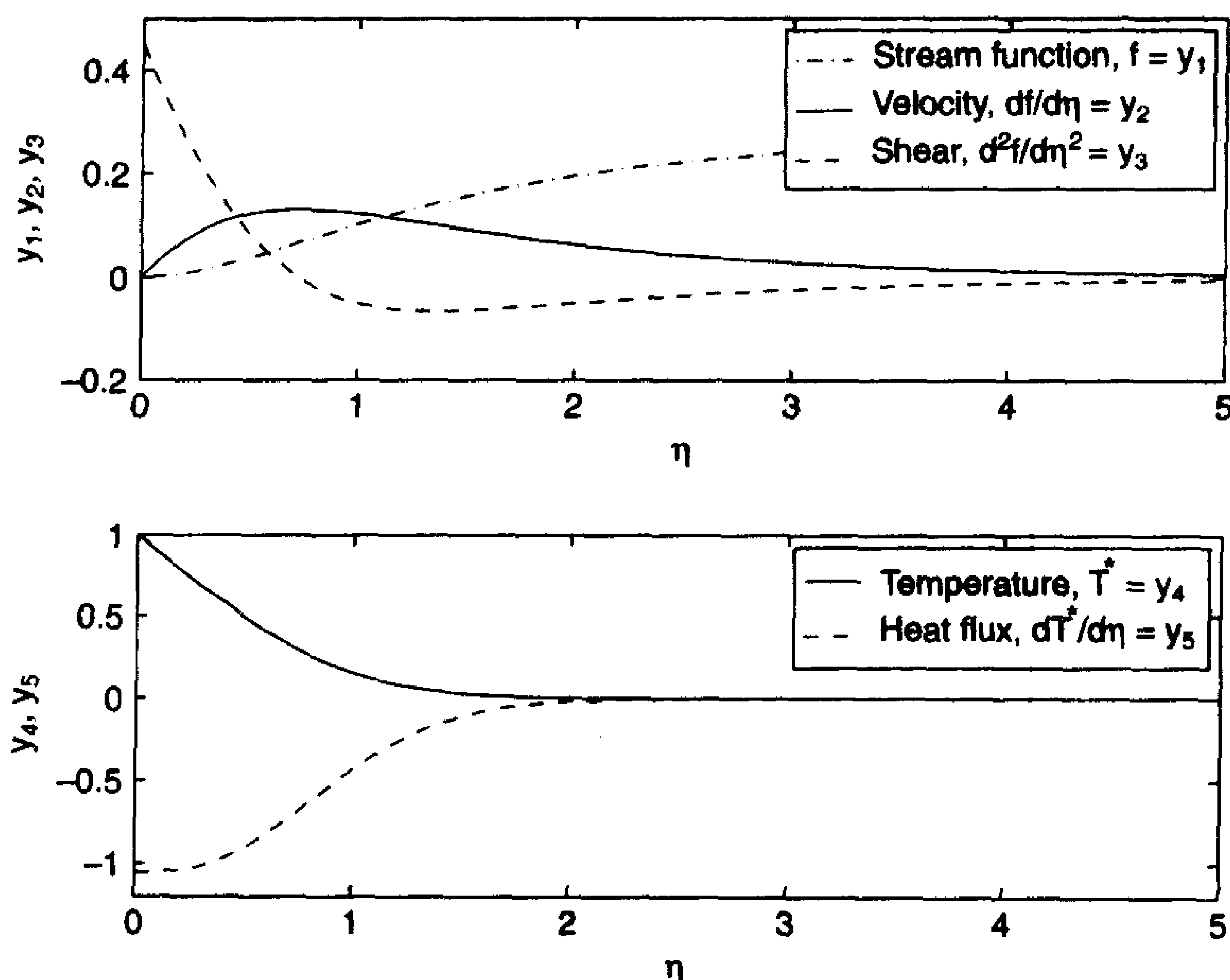


图 12.15　当 Pr = 7.0 时的自然对流的解

流函数的最大值反映了自然对流的泵效应，与 Pr 直接相关。Pr 的较大值对应于修正流函数 f 的较小值，修正流函数的最大值与总体积流量有关。要解释给定流体的这种关系，必须对一维流函数进行计算：

$$\Psi(x,y) = 4f(\eta)v\left(\frac{Gr_x}{4}\right)^{0.25}$$

通过计算可以看出，在相同的温差下，气体的体积流量要大于水的体积流量。

由于这种流动可以将能量带离物体表面，所以对于热流量的分析可以采用相同的方法，由下式确定：

$$q''_s = -\frac{k(T_s - T_\infty)}{x}\left(\frac{Gr_x}{4}\right)^{0.25}\frac{\mathrm{d}T^*}{\mathrm{d}\eta}\bigg|_{\eta=0}$$

计算出气体和水的热流量后，可以发现水的热流量比空气大 100 倍。主要是由热导、比热和密度共同决定热流量的大小。在温度为 300 K 的常压下，水的热导大约是空气的 30 倍，比热为空气的 4 倍，密度为空气的 1000 倍。虽然在空气中体积流量大，但是在水中热流量大。

12.3.3　印刷电路板的温度分布①

印刷电路板如图 12.16 所示，其上有 4 个平面固定的电子器件，发散出的能量大小在图中给出。电路板 y 轴垂直安装在支架上，在这种情况下传热系数 h 在 y 方向上是变化的。为了简化，假设在 x 方向上传热系数为一常数。为了近似描述 y 方向上传热系数的变化情况，在这里可以认为该板是由三块相邻的大小相等的板组成，三块板的传热系数各不相同，如

① 由 Yogendra Joshi, Department of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 提出。

图 12.16所示。板和器件的热导 k 相同且与所在位置无关。为简单起见,假设四个器件具有相同的热导系数。

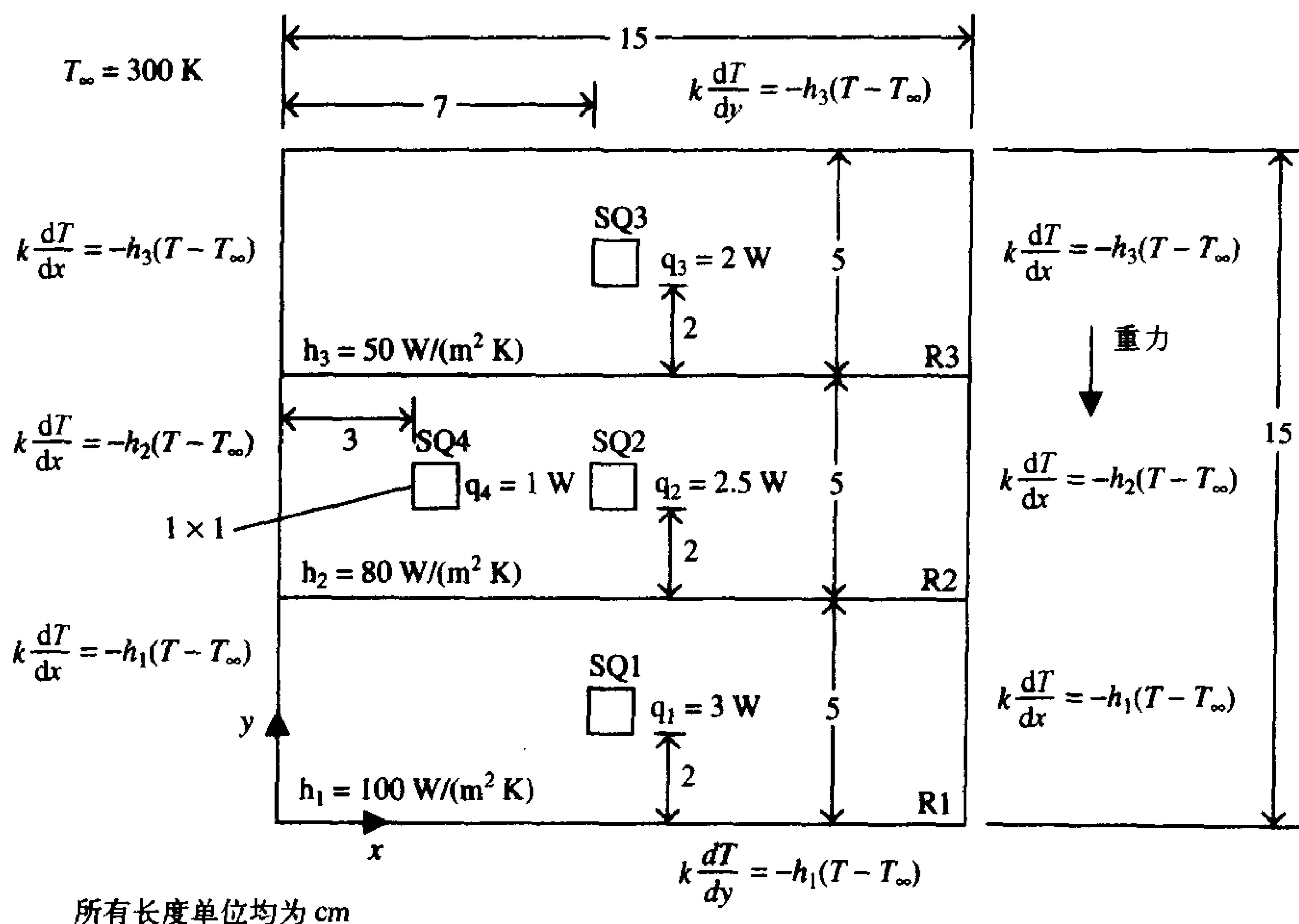


图 12.16　印刷电路板的几何尺寸和参数值

每块板和电子器件特征方程为:

$$\nabla(k\nabla T) + q''' - H(T - T_\infty) = 0$$

其中,q'''是每单位体积热源的功率(W/m³),$H = 2h/t$(W/m³K),t 是板的厚度,T_∞是环境温度。由于板的两面都有热对流发生,所以在 H 的定义式中有系数 2。

例 12.7　印刷电路板的温度和流量的确定

采用 PDE 工具箱可以计算整块板的温度分布。首先,创建一个适当的绘图环境。进入 *Options* 菜单,设置 *Axis Limits* 为[0,17],两个轴的格间距为 1,然后选择 *Snap*。下一步,将 *Generic Scalar* 变为 *Heat Transfer*。然后画出三个矩形和四个正方形,尺寸和位置如图 12.16 所示。结果生成设置表达式 $R1 + R2 + R3 + SQ1 + SQ2 + SQ3 + SQ4$。每个区域将由导热方程确定。然而,必须确定表 12.6 所示矩形区域的边界条件。因为每个 SQn 全部存在于提前定义的区域(Rn)内,对于 PDE 工具而言并不产生新的边界。所以,设置表达式中的加号(+)并不具有创建边界的功能。注意,k, h 和 Q 的单位分别为 W/(cm K), W/(cm²K)和 W/(cm³),与板的尺寸单位一致。

再下一步,进入 *Boundary Mode*,然后选择 *Specify Boundary Conditions*。表 12.6 给出的矩形边界条件是纽曼类型,这些边界都用带箭头的红线标出。双击每条线后,表 12.6 给出的值会输入到相应的位置。然后选择 *PDE Mode*,将光标移到七个区域中的某一区域,并双击,显示 *PDE Specification* 窗口,表 12.6 给出的数值可通过该窗口输入。

对网格初始化并优化,然后得出结果。绘制出温度分布图和热流量图,如图 12.17 和图 12.18所示。

表 12.6 印刷电路板的数据以及函数 pdetool 所使用的符号

	区域						
	*R*1	*R*2	*R*3	*SQ*1	*SQ*2	*SQ*3	*SQ*4
边界条件							
左							
$g \to hT_\infty$ (W/cm^2)	3	2.4	1.5	—	—	—	—
$q \to h$ (W/cm^2 K)	0.01	0.008	0.005	—	—	—	—
右							
$g \to hT_\infty$ (W/cm^2)	3	2.4	1.5	—	—	—	—
$q \to h$ (W/cm^2 K)	0.01	0.008	0.005	—	—	—	—
顶							
$g \to hT_\infty$ (W/cm^2)	—	—	1.5	—	—	—	—
$q \to h$ (W/cm^2 K)	—	—	0.005	—	—	—	—
底							
$g \to hT_\infty$ (W/cm^2)	3	—	—	—	—	—	—
$q \to h$ (W/cm^2 K)	0.01	—	—	—	—	—	—
PDE 说明							
$k \to k$ (W/cm K)	0.003	0.003	0.003	0.003	0.003	0.003	0.003
$Q \to Q$ (W/cm^3)	0	0	0	30	25	20	10
$h \to H$ (W/cm^3 K)	0.1	0.08	0.05	0.1	0.08	0.05	0.08
Text $\to T_\infty$ (K)	300	300	300	300	300	300	300

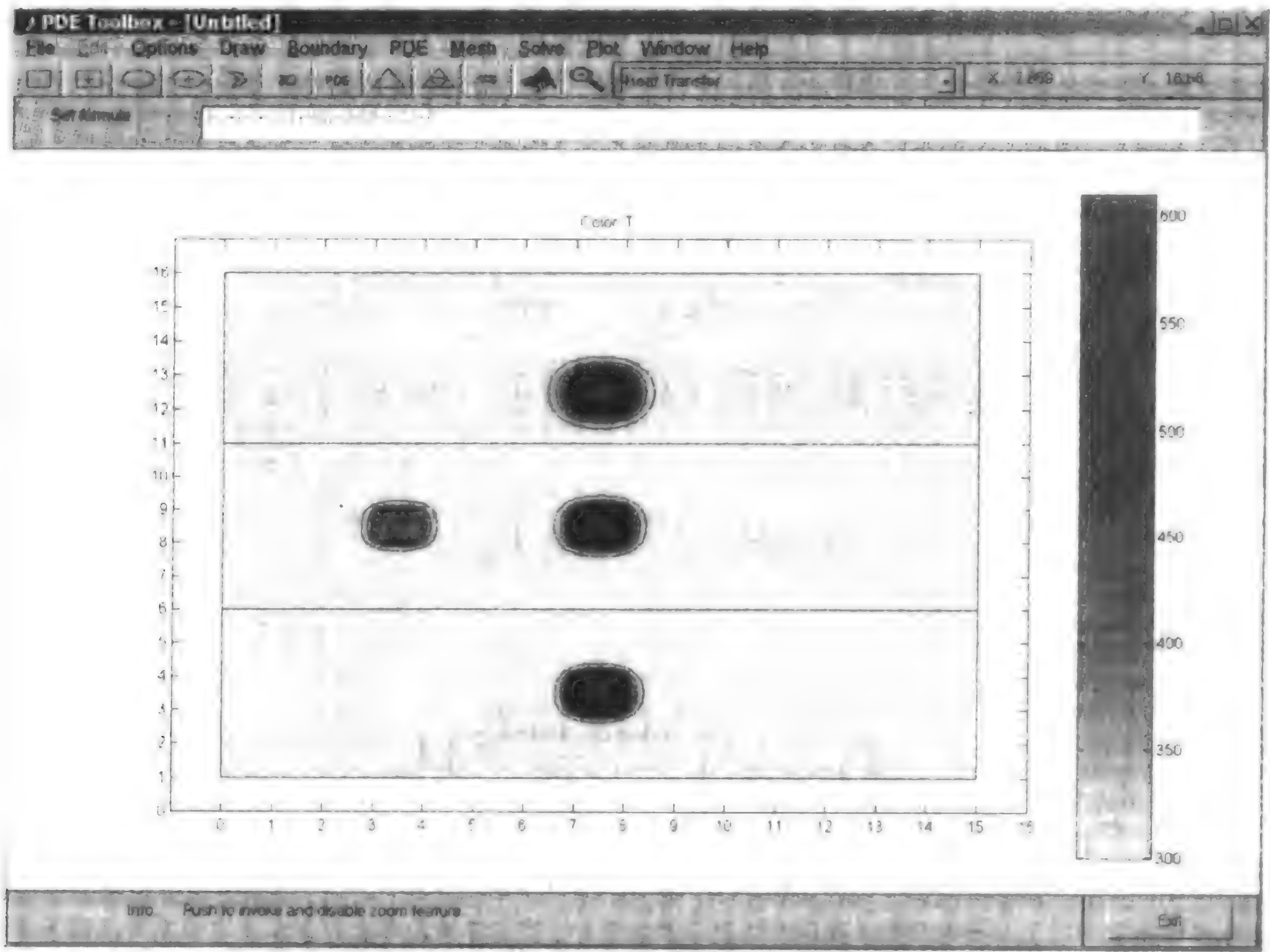


图 12.17 图 12.16 所示印刷电路板的温度分布

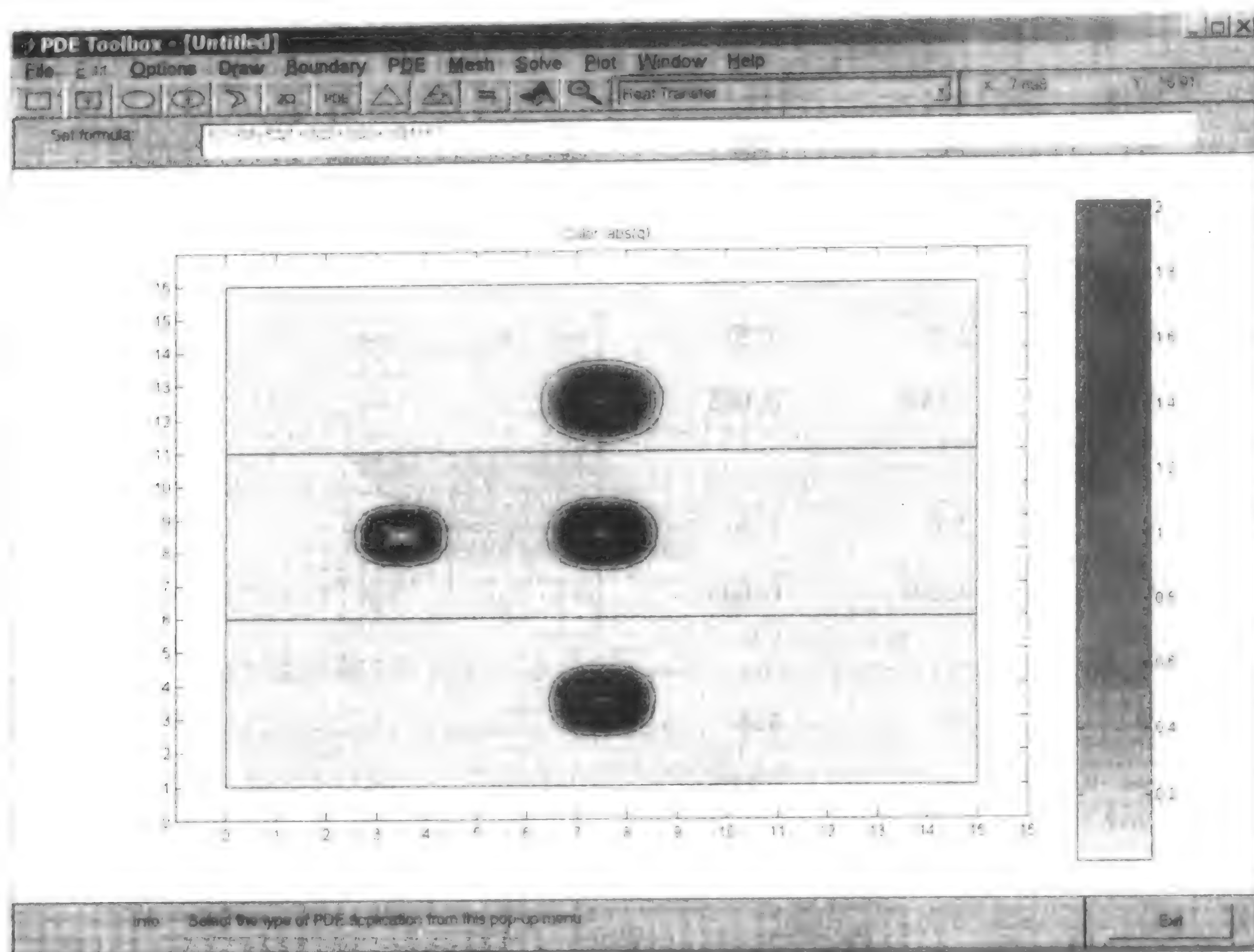


图 12.18　图 12.16 所示印刷电路板的热流量

12.4　辐射传热

12.4.1　辐射角系数——平行平面上的面积微元到任意矩形

分析扩散表面包壳的热辐射时需要计算辐射角系数。在许多计算角系数的方法中，较常用的是根据角系数的定义式对其进行求解。首先，考虑面积微元 dA_1 和一个有限面积 A_2 间角系数的一般表达式①：

$$dF_{2-d_1} = \frac{dA_1}{A_2}\int_{A_2}\frac{\cos\theta_1\cos\theta_2}{\pi S^2}dA_2 \tag{12.2}$$

式中，S 为 dA_1 到 A_2 上某一点的距离，如图 12.19 所示。角 θ_j，$j = 1, 2$，是 S 与表面法线的夹角。角系数互换性为：

$$A_2 dF_{2-d_1} = dA_1 F_{d_1-2} \tag{12.3}$$

因此，F_{d_1-2} 可以表示为：

$$F_{d_1-2} = \int_{A_2}\frac{\cos\theta_1\cos\theta_2}{\pi S^2}dA_2 \tag{12.4}$$

然后进行数值积分。假设 dA_1 和 A_2 相互平行且 A_2 是一个矩形，通过适当的程序设计可以将这些限制条件去掉。在这个例子中，方程式(12.4)可以改写为：

---

① 参见 R. Siegel 和 J. R. Howell, *Thermal Radiation Heat Transfer*，第三版，Hemisphere Pub.，Washington，DC，1992 年，第 189～252 页。

$$F_{d_1-2} = \frac{1}{\pi}\int_{y_{2a}}^{y_{2b}}\int_{x_{2a}}^{x_{2b}} \frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}x_2\mathrm{d}y_2$$

式中,向量 $\boldsymbol{S}$ 为:

$$\boldsymbol{S} = \boldsymbol{x}_2 - \boldsymbol{x}_1 = (x_2 - x_1)\boldsymbol{i} + (y_2 - y_1)\boldsymbol{j} + (z_2 - z_1)\boldsymbol{k}$$

和

$$S = |\boldsymbol{S}|$$

角度可以由向量 $\boldsymbol{S}$ 和矩形上的法线向量 $\boldsymbol{n} = \boldsymbol{k}$ 表示:

$$\cos\theta_1 = \cos\theta_2 = \frac{\boldsymbol{n}\cdot\boldsymbol{S}}{|\boldsymbol{S}|}$$

由于两个平面相互平行,式中 $\theta_1 = \theta_2 = \theta$。所以:

$$F_{d_1-2} = \frac{1}{\pi}\int_{y_{2a}}^{y_{2b}} I_{x2}(y_2)\mathrm{d}y_2 \tag{12.5}$$

式中,

$$I_{x2}(y_2) = \int_{x_{2a}}^{x_{2b}} f(x_2, y_2)\mathrm{d}x_2$$

且

$$f(x_2, y_2) = \frac{\cos^2\theta}{|\boldsymbol{S}|^2} = \frac{(\boldsymbol{n}\cdot\boldsymbol{S})^2}{|\boldsymbol{S}|^4}$$

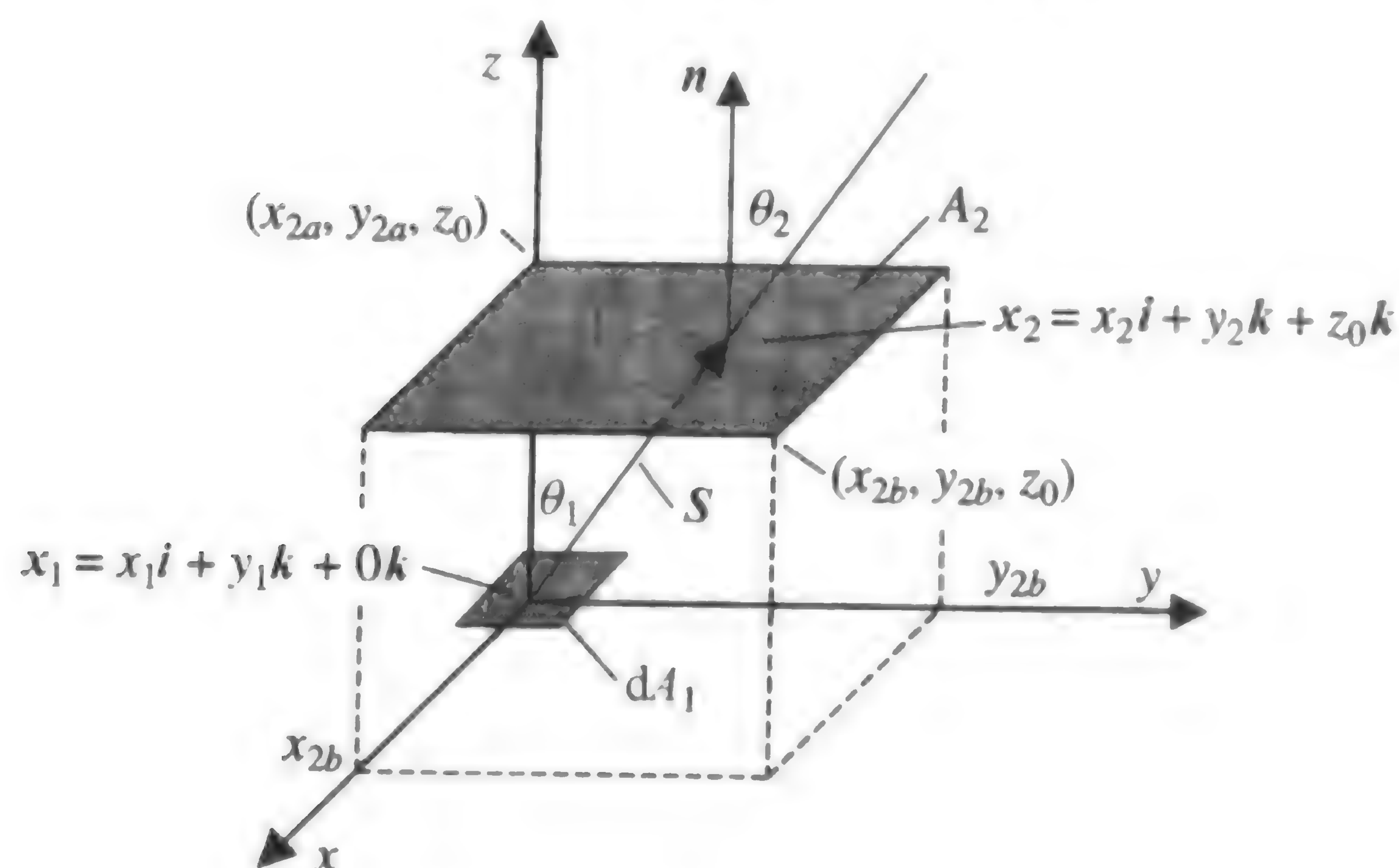


图 12.19 面积微元和有限矩形平行分布的几何图形

例 12.8 平行板间面积微元和有限矩形的角系数

为说明式(12.5)的数值积分,研究表 12.7 中的两组数据。在第一个例子中,得到如表最后一行所示的角系数。在第二个例子中,给出了表中组 # 1 的角系数随表面距离变化的曲线。

创建主函数 **Fd1_2** 和子函数 **kernel2** 来计算 A_2 上任意一点 $f(x_2, y_2)$。下面给出了这些函数。该函数用向量形式确定长度 S 和 $\cos\theta$。注意,为了与函数 `dblquad` 保持一致,**kernel2** 函数必须返回一个向量,该向量的维数必须与输入向量 x 的维数相等。这样函数 `dblquad` 既能减少被积函数调用次数,同时又能提供所需的数据。**kernel2** 函数通过用 `length` 确定 x 元素个数来实现此项功能:

```
function F=Fd1_2(x_2a,x_2b,y_2a,y_2b,dz)
F=dblquad(@kernel2,x_2a,x_2b,y_2a,y_2b,[],[],dz)/pi;

function f=kernel2(x,y,dist)
L=length(x);
S=[x;linspace(y,y,length(x));dist*ones(1,length(x))];
n=repmat([0,0,1]',1,L);
f=dot (n,S).^2./dot(S,S).^2;
```

表 12.7　实例的几何尺寸与角系数值

参数	组#1	组#2
A_2 的几何尺寸		
原点的 x 坐标 x_{2a}	-1	-1
原点的 y 坐标 y_{2a}	-1	-1
顶点的 x 坐标 x_{2b}	0	1
顶点的 y 坐标 y_{2b}	0	1
两平面间的距离 z_0	5	1
角系数的计算值 F_{d_1-2}	0.0121	0.5541

利用这些函数计算表 12.7 中的两组数据。程序为:

```
Set1=Fd1_2(-1,0,-1,0,5)
Set2=Fd1_2(-1,1,-1,1,1)
```

执行结果如表 12.7 中的最后一行所示。

绘制表 12.7 中组#1 的角系数随距离变化的曲线,程序为:

```
x_2a=0;x_2b=1;y_2a=0;y_2b=1;
dz=linspace(0.1,5,100);
for i=1:length(dz)
 Fd12(i)=Fd1_2(x_2a,x_2b,y_2a,y_2b,dz(i));
end
plot(dz,Fd12,'k-')
xlabel('Separation distance of surfaces')
Ylabel('View factor')
```

执行结果如图 12.20 所示。当两个平行平面距离趋于 0 时,F_{d_1-2}趋于极限值 0.25。这是因为点 dA_1 与 A_2 的一个顶角在同一条直线上,所以当这两个平行平面相互靠近时 A_2 切掉了 dA_1 的 1/4。在组#2 的几何尺寸中,dA_1 和 A_2 的中心点在同一直线上,当两个平行平面相互靠近时,F_{d_1-2}的极限值为 1.0。

12.4.2　平行平面上两个矩形间的角系数

先前计算的是从面积微元到有限面积之间的角系数。通过对面积微元积分,可以得到两个面积为有限的平面间的角系数。角系数由下式定义:

$$F_{2-1}=\frac{1}{\pi A_2}\int\limits_{A_1}\int\limits_{A_2}\frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}A_2\mathrm{d}A_1$$

$$=\frac{1}{\pi A_2}\int_{y_{1a}}^{y_{1b}}\int_{x_{1a}}^{x_{1b}}\int_{y_{2a}}^{y_{2b}}\int_{x_{2a}}^{x_{2b}}\frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}x_2\mathrm{d}y_2\mathrm{d}x_1\mathrm{d}y_1 \tag{12.6}$$

式中的变量定义与前面一节所讨论的变量定义相似。图 12.21 定义了式中四重积分的积分限。有许多方法可以计算四重积分，在此使用直接积分法。两个平面平行，其边缘也平行，式(12.6)可以写为：

$$F_{2-1} = \frac{1}{\pi A_2}\int_{y_{1a}}^{y_{1b}} I_{x1}(y_1)\mathrm{d}y_1$$

其中，

$$I_{x1}(y_1) = \int_{x_{1a}}^{x_{1b}} I_{y2}(x_1, y_1)\mathrm{d}x_1$$

$$I_{y_2}(x_1, y_1) = \int_{y_{2a}}^{y_{2b}} I_{x2}(x_1, y_1, y_2)\mathrm{d}y_2$$

$$I_{x2}(x_1, y_1, y_2) = \int_{x_{2a}}^{x_{2b}} f(x_1, y_1, x_2, y_2)\mathrm{d}x_2$$

$$f(x_1, y_1, x_2, y_2) = \frac{\cos^2\theta}{|\boldsymbol{S}|^2} = \frac{(\boldsymbol{n}\cdot\boldsymbol{S})^2}{|\boldsymbol{S}|^4}$$

$$\boldsymbol{S} = (x_1 - x_2)\boldsymbol{i} + (y_1 - y_2)\boldsymbol{j} + (z_1 - z_2)\boldsymbol{k}$$

并且，$\boldsymbol{n} = \boldsymbol{k}, z_1 = 0, z_2 = z_0$。

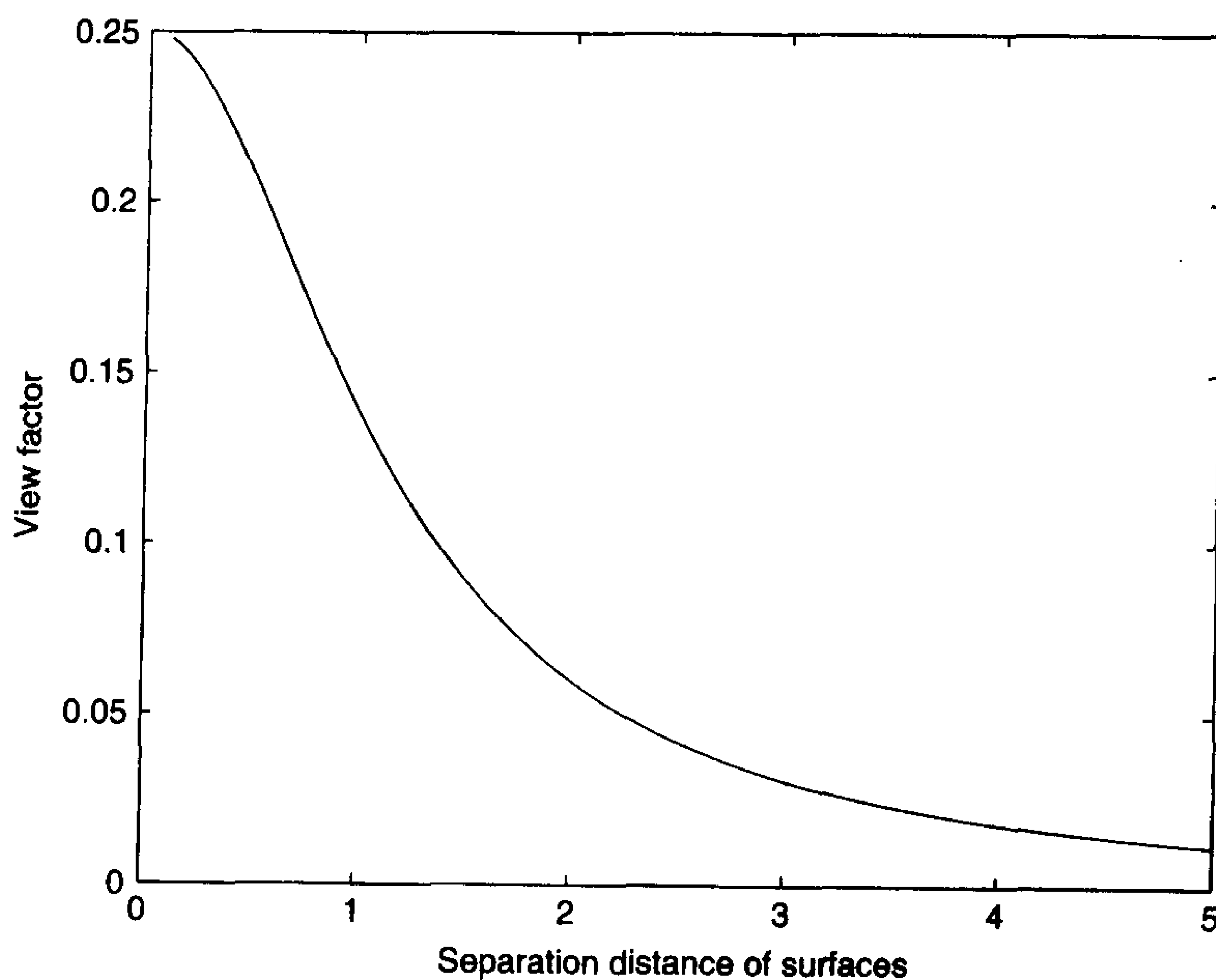


图 12.20　图 12.19 所示两个平行平面间的距离与角系数的函数关系。面积在平行板间，面积微元和有限面积的一角在同一直线上

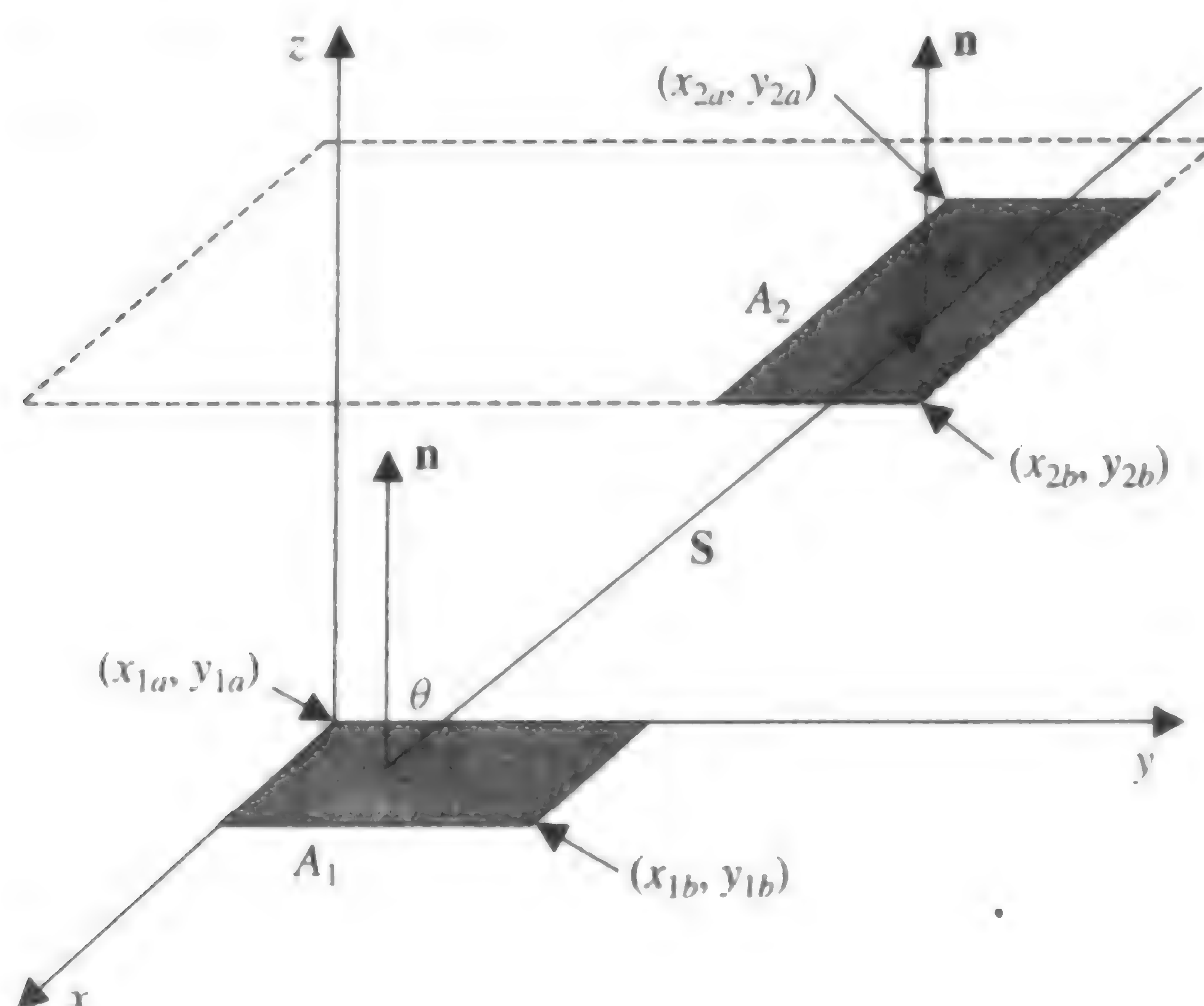


图 12.21 确定两个平行矩形之间角系数的几何图形

例 12.9 两个平行矩形间的角系数

计算表 12.8 中两组数据的角系数。使用的方法是嵌套两次调用 dblquad。这个功能是在函数 **F1_2** 中实现的,包括两个核子函数。函数 **InnerKernel** 用于计算 $f(x_1, y_1, x_2, y_2)$,函数 **OuterKernel** 用于计算 $I_{y2}(x_1, y_1)$,也就是计算在 A_2 上对 A_1 的点的积分。在主函数中使用 dblquad 完成 A_1 的所有点的积分。

```
function F1_2(x1a,x1b,y1a,y1b,x2a,x2b,y2a,y2b,dz)
F12 = dblquad(@QuterKernel,x_1a,x1b,y1a,y1b,[],[],x2a,x2b,y2a,y2b,...dz)/(A2 * pi)
function f = QuterKernel(x1,y1,x2a,x2b,y2a,y2b,dz)
for i = 1:length(x1)
  f(i) = dblquad(@InnerKernel,x2a,x2b,y2a,y2b,[],[],dz,x1(i),y1);
end

function f = InnerKernel(x,y,dz,x2,y2)
L = length(x);
S = [x-x2 * ones(1,L);(y-y2) * ones(1,L);dz * ones(1,L)];
n = repmat([0,0,1]',1,L);
f = dot(n,S).^2./dot(S,S).^2;
```

表 12.8 计算角系数所需数据

参数	组 # 1	组 # 2
A_1 的几何尺寸		
原点的 x 坐标,面积 1, x_{1a}	-1	-2
原点的 y 坐标,面积 1, y_{1a}	-1	-2
顶点的 x 坐标,面积 1, x_{1b}	1	0
顶点的 y 坐标,面积 1, y_{1b}	1	0
A_2 的几何尺寸		
原点的 x 坐标,面积 2, x_{2a}	-1	2
原点的 y 坐标,面积 2, y_{2a}	-1	2
顶点的 x 坐标,面积 2, x_{2b}	1	0
顶点的 y 坐标,面积 2, y_{2b}	1	0
平面间的距离	2	2
角系数计算值(F_{2-1})	0.1998	0.0433

然后就可以计算表 12.8 中两组数据的平行平面上任意矩形的角系数。程序为：

```
Set1 = F1_2( - 1,1, - 1,1, - 1,1, - 1,1,2)
Set2 = F1_2( - 2,0, - 2,0,2,0,2,0,2)
```

执行以上程序可以得到如表 12.8 中最后一行所示的结果。对两组数据来说，两矩形之间的面积和平行板间的空间是相同的。但是，组 #1 中的矩形是正好相反的，而组 #2 中的矩形是互相抵消的。通过比较正好相反矩形例子的有效分析解可以验证表 12.8 中组 #1 的结果①，也可以发现这个结果有四位有效数字。

12.4.3 具有漫射、灰体壁面的包壳辐射

考虑热辐射时，经常遇到的问题是要根据包壳的辐射来确定温度和传热率，该包壳具有漫射和灰度表面而且包含一种不参加辐射的媒介物。这种情况发生在炉子、房间和其他被包围起来的空间里。采用漫射灰体表面的假设，使其和一般辐射模式相比具有较小的复杂性。漫射说明所有表面辐射和吸收强度与方向无关，灰度说明发射率和吸收率与波长无关。然而，即使这样简化，包壳的问题仍需要用相当大的精力去加以解决。这些问题可以很自然地在矩阵中表现出来，因此，MATLAB 提供了一个理想的环境对其进行系统阐述并予以解决。由该分析得出下列公式②：

$$\frac{Q_k}{A_k} = q_k = \frac{\varepsilon_k}{1-\varepsilon_k}(\sigma T_k^4 - q_{0,k}) \tag{12.7}$$

$$\frac{Q_k}{A_k} = q_k = q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} = \sum_{j=1}^{N} F_{k-j}(q_{0,k} - q_{0,j}) \tag{12.8}$$

其中，Q_k 是表面 k 的传热率。A_k 为面积，q_k 是热流量，$q_{0,k}$ 是辐射系数，F_{k-j} 为从 k 表面辐射出去而被 j 表面接收的能量的角系数，N 为包壳的表面个数，$\sigma = 5.67\times10^{-8}$ W/(m² K⁴)是斯蒂芬－玻耳兹曼(Stefan-Boltzmann)常数。该公式假设每个表面接收和发射的辐射作为一整体，其强度与方向无关。对于一个给定的问题，可以通过对包壳表面进行细分直到获得的结果与面积分割无关，从而对该假设进行计算。

对于一般的包壳问题，必须指定每个表面的传热率或温度。一旦指明后，式(12.7)和式(12.8)就确定了每个表面惟一独立的关系。如果指定了温度，则式(12.7)和式(12.8)相等，表示为：

$$q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} = \frac{\varepsilon_k}{1-\varepsilon_k}(\sigma T_k^4 - q_{0,k}) \tag{12.9}$$

如果指定了传热率，则式(12.9)可以改为：

$$\frac{Q_k}{A_k} = q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} \tag{12.10}$$

写成矩阵形式：

$$\begin{bmatrix} d_1 - F_{1-1} & -F_{1-2} & \cdots & -F_{1-N} \\ -F_{2-1} & d_2 - F_{2-2} & & -F_{1-2} \\ \vdots & & & \vdots \\ -F_{N-1} & -F_{N-2} & & d_N - F_{N-N} \end{bmatrix} \begin{bmatrix} q_{0,1} \\ q_{0,2} \\ \vdots \\ q_{0,N} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

① 参见 R.Siegel 和 J.R.Howell，出处同上，第 1030 页。

② 参见 R.Siegel 和 J.R.Howell，出处同上，第 189～252 页。

当指定了温度时,式中,

$$d_k = 1/(1-\varepsilon_k)$$

$$b_k = \frac{\varepsilon_k}{1-\varepsilon_k}\sigma T_k^4$$

当指定了传热率时,式中,

$$d_k = 1$$

$$b_k = \frac{Q_k}{A_k}$$

矩阵中每一行代表一个表面,其形式取决于该表面是指定温度还是指定传热率。对以上任何一种情况,表面的辐射系数都是未知的。因此,方程组中包含 N 个方程和 N 个未知辐射系数。一旦辐射系数被确定,对于给定表面未知的温度和传热率就可由式(12.7)和式(12.8)确定。除特殊情况外,用式(12.7)计算很简单,当发射率为 1 时,应使用式(12.8)。

例 12.10　矩形包壳的总传热率

对于一个横截面为矩形的炉子,假设可无限延伸到纸面内部。图 12.22 定义了其几何尺寸。表 12.9 定义了其角系数,这些系数可以通过霍特尔(Hottel)的 crossed-string 方法计算①。表中粗体印刷的值是待选值,在此基础上,对角系数进行代数运算求出其他值。

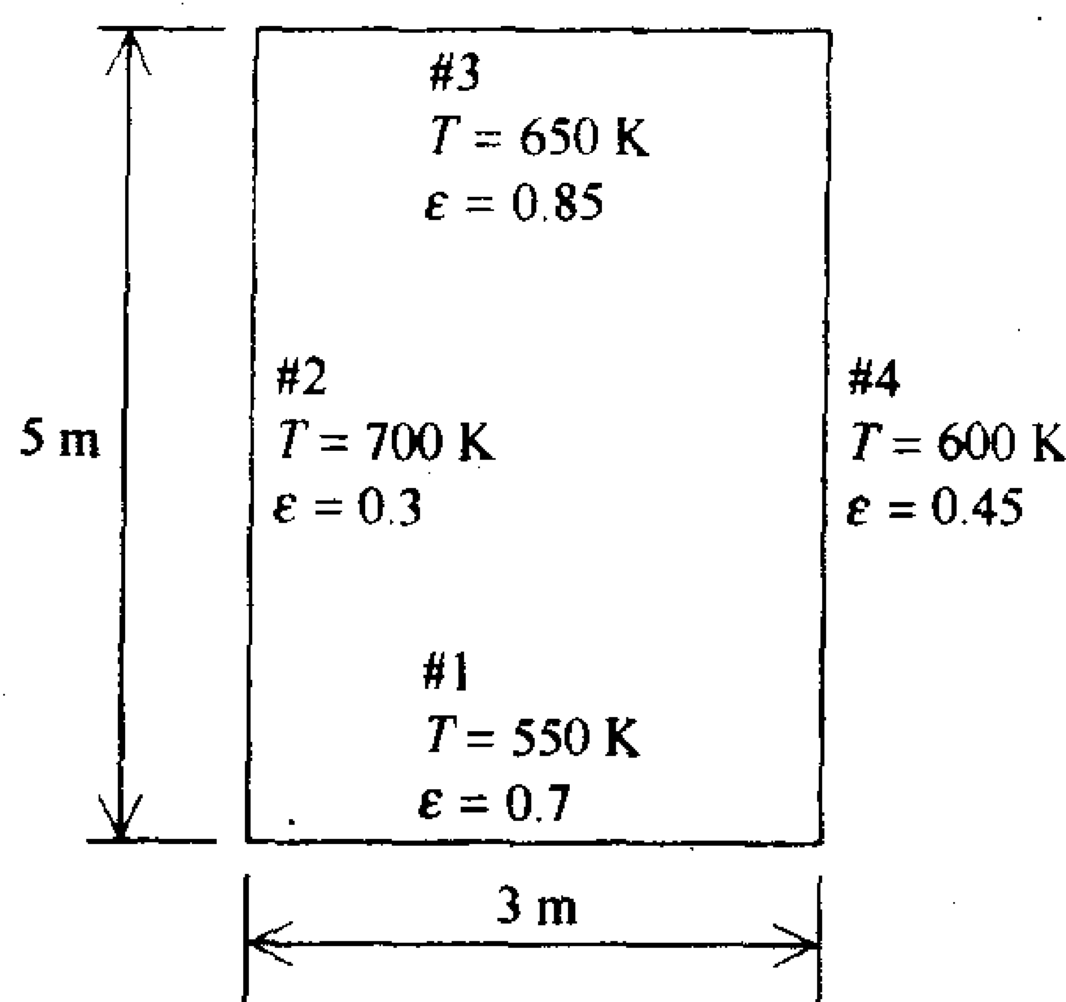


图 12.22　具有灰度侧壁的包壳辐射时的几何尺寸和表面特性

表 12.9　图 12.22 所示包壳的角系数 F_{i-j}

i/j	1	2	3	4
1	**0**	**0.3615**	**0.2770**	0.3615
2	0.2169	**0**	**0.2169**	0.5662
3	0.2770	0.3615	**0**	0.3615
4	0.2169	0.5662	0.2169	0

下面的程序利用了标识符 $c_k, k = 1, 2, \cdots, N$,当给定温度时,$c_k = 0$;当给定传热率时,$c_k = 1$。然后根据 c_k 选择 d_k 和 b_k 的值。此外,所有的向量维数必须为 N 且矩阵 F 为 $(N \times N)$阶。程序为:

```
sigma = 5.6693e - 8;N = 4;
A = [3,5,3,5];epsilon = [0.7,0.3,0.85,0.45];
```

① 参见 R.Siegel 和 J.R.Howell, 出处同上。

```
T = [550,700,650,600];
F = -[0,0.3615,0.277,0.3615;...
      0.2169,0,0.2169,0.5662;...
      0.277,0.3615,0,0.3615;...
      0.2169,0.5662,0.2169,0];
Q = [0,0,0,0];
c = [0,0,0,0];
b = sigma * epsilon./(1 - epsilon).*(1 - c).*T.^4 + c.*Q./A;
d = (1 - c).*1./(1 - epsilon) + c;
for k = 1:N
 F(k,k) = d(k) + F(k,k);
end
q0 = F \ b';
Q = A.*epsilon./(1 - epsilon).*(1 - c).*(sigma * T.^4 - q0')
q = Q./A
```

执行程序后得到：

$Q = [-8627.9\ 8061.1\ 4525.9\ -3959.1]\mathrm{W}, q = [-2876\ 1612.2\ 1508.6\ -791.8]\ \mathrm{W/m^2}$。

可以看出，传热率 Q 之和接近于零。

12.4.4 炉内热平板的瞬态热辐射[①]

假设有一垂直悬挂在炉内的平板。炉子的一内壁与含有热元件的平板表面平行，炉子和平板的温度最初都为室温。为了使平板的温度在 t_h 时间内上升为 T_e，加热元件所需的燃烧热能 Q 可由平板和炉壁内能量平衡关系确定，从而得出下式：

$$\frac{\mathrm{d}T_w}{\mathrm{d}t} = P_1 Q - P_2(T_w^4 - T_p^4)$$

$$\frac{\mathrm{d}T_p}{\mathrm{d}t} = -P_3(T_p^4 - T_w^4)$$

其中，平板和炉内壁作为一个整体考虑，T_w 是炉内壁的温度，T_p 是平板的温度。如果将这种结构假设为具有两个漫射灰度表面的包壳，则[②]：

$$P_1 = \frac{1}{m_w c_w}$$

$$P_2 = \frac{\sigma}{m_w c_w}\left[\frac{1-\varepsilon_p}{\varepsilon_p A_p} + \frac{1}{A_w F_{wp}} + \frac{1-\varepsilon_w}{\varepsilon_w A_w}\right]^{-1}$$

$$P_3 = \frac{\sigma}{m_p c_p}\left[\frac{1-\varepsilon_p}{\varepsilon_p A_p} + \frac{1}{A_p + F_{pw}} + \frac{1-\varepsilon_w}{\varepsilon_w A_w}\right]^{-1}$$

其中，m_p 和 m_w 分别为平板和炉内壁的质量；c_p 和 c_w 分别为平板和炉内壁的比热；ε_p 和 ε_w 分别为平板和炉内壁的辐射率，A_p 和 A_w 分别为平板和炉内壁的面积；F_{pw} 和 F_{wp} 分别为平板和炉内壁的角系数；$\sigma = 5.67 \times 10^{-8}\ \mathrm{W/(m^2\ K^4)}$ 是斯蒂芬-玻耳兹曼常数。

例 12.11 炉内平板的瞬态热辐射

对于某一结构，假设 $P_1 = 1.67 \times 10^{-5}\ \mathrm{K/J}$，$P_2 = 8.8 \times 10^{-14}\ \mathrm{s^{-1}K^{-3}}$ 和 $P_3 = 6.3 \times$

---

① 由 Yogendra Joshi, Department of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 提出。

② 参见 F.P.Incropera 和 D.P.DeWitt，出处同上，第 13 章。

10^{-13} $s^{-1}K^{-3}$。然后，确定使平板温度在时间 t_h = 10 min(600 s)内上升到 T_e = 1100 K 所需的 Q 值。假设平板和炉子的初始值为 300 K。

首先，创建一个主函数 **TranRadHeat** 和两个子函数，一个由函数 ode45 用来求解两个一阶微分方程，称为 **RadTemp** 函数；另一个由函数 fzero 用来确定 Q 值，称为 **Qgen** 函数。对于函数 **Qgen** 有：T1o = $T_w(0)$，T2o = $T_p(0)$，用函数 interp1 来确定温度 T_e = Te = 1100 K 时的时间。

```
function TranRadHeat
global Te th P1 P2 P3 T1o T2o tend
P1 = 1.67e-5;P2 = 8.8e-14;P3 = 6.3e-13;
Qguess = 100000;Te = 1100;th = 600;tend = 660;
T1o = 300;T2o = 300;
options = optimset('display','off');
Q = fzero(@Qgen,Qguess,options);
[t,T] = ode45(@RadTemp,[0,tend],[T1o;T2o],[],Q);
plot(t,T(:,1),'k-',t,T(:,2),'k--')
z = axis;
hold on
plot([0,z(2)],[Te,Te],'k',[th,th],[z(3),z(4)],'k')
xlabel('Time(s)')
text(0.05*z(2),0.85*z(4),['Q='num2str(Q,6)'W'])
ylabel('Temperature(K)')
legend('Wall temperature','Plate temperature','Location','NorthWest')

function dTdt = RadTemp(t,T)
dTdt = [P1*Q-P2*(T(1)^4-T(2)^4);-P3*(T(2)^4-T(1)^4)];

function PlateTempDev = QGen(Q)
global Te th P1 P2 P3 T1o T2o tend
[t,T] = ode45(@RadTemp,[0,tend],[T1o;T2o],[],Q);
PlateTempDev = Te - interp1(t,T(:,2),th,'spline');
```

程序执行后产生图 12.23。

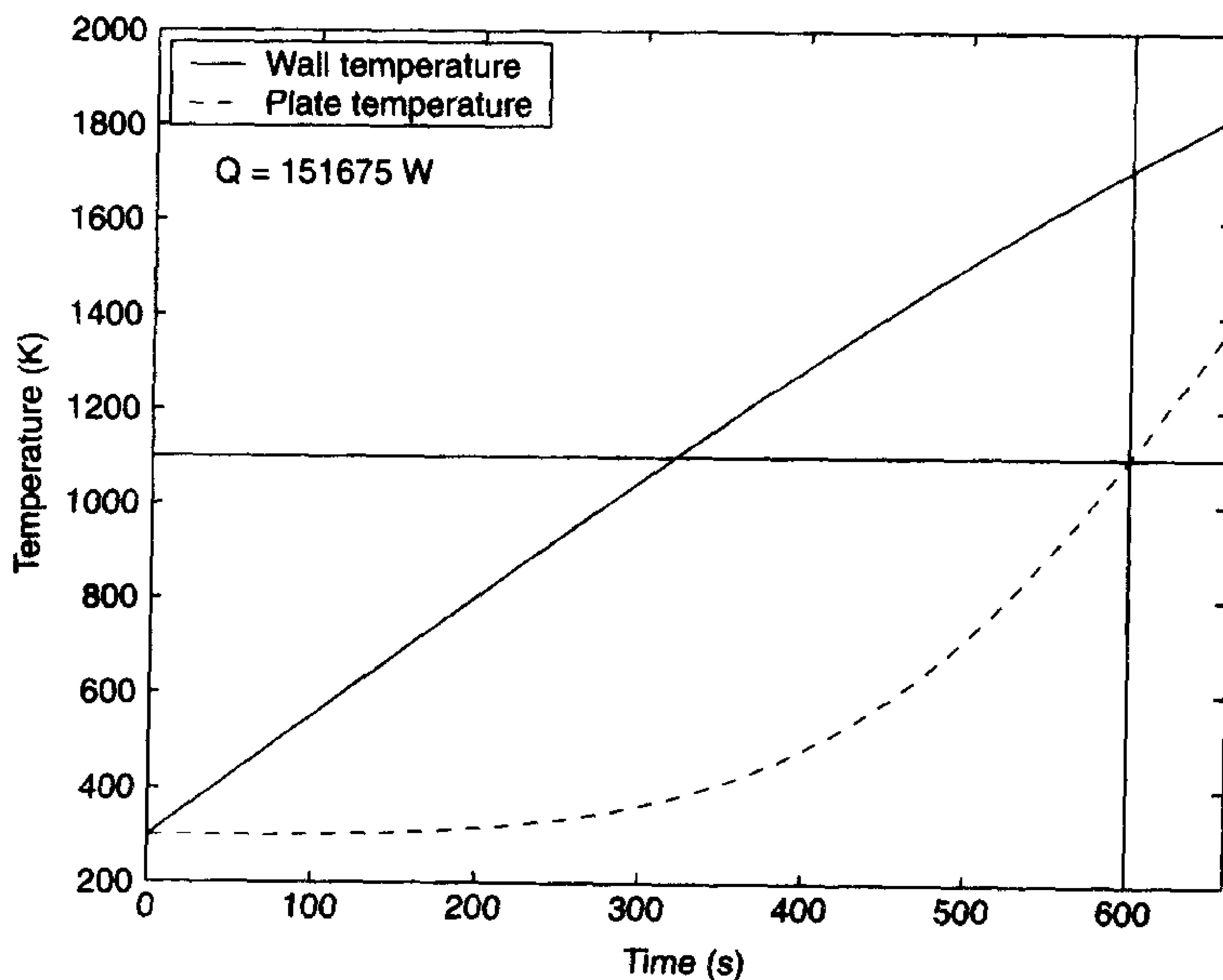


图 12.23 平板和炉壁温度随时间变化的曲线图

练习

12.1 一维大平壁导热可以表示为：

$$-k(T)\frac{\mathrm{d}T}{\mathrm{d}x}=q$$

其中，q 为热流量，$k(T)$是取决于温度的导热系数，T 是温度，x 是空间坐标。设侧壁由绝缘矿渣绒构成，导热系数根据下式变化：

$$k(T)=-k_o+k_s \qquad 240<T<365\ \mathrm{K}$$

其中，$k_0=0.48, k_s=0.000\,32$，k 的单位为 W/m K，T 的单位为 K。当热流量 $q=12.5\ \mathrm{W/m^2}$，侧壁厚 0.1 m，并且 $x=0.1$ m 处侧壁的表面温度为 300 K 时，确定 $x=0.1$ m的温度。计算结果与下式做比较：

$$k_0(T-T_1)+\frac{k_s}{2}(T^2-T_L^2)=q(x-L)$$

其中，$T(0)=T_L$。根据导热系数的平均值计算热流量，结果与真实值非常接近。

12.2 为了估计加热(冷却)牛奶的时间，一个标准的塑料牛奶瓶可以看做一个集总热容。这种条件下的特征方程可以表示为：

$$\frac{\mathrm{d}T}{\mathrm{d}t}=\frac{Q}{mc_v}$$

式中，$Q=hA(T_{amb}-T)$是周围环境向牛奶瓶进行的热传递，m 是瓶子的质量，c_v 是瓶子的比热。对于一个简单的辐射模型为：

$$Q=A\sigma\varepsilon(T^4-T_{amb}^4)$$

式中，$\sigma=5.667\times10^{-8}\ \mathrm{W/m^2\ K^4}$ 是斯蒂芬－玻耳兹曼常数，ε 是发射率，A 是表面积，T 是瓶子的温度，T_{amb}是环境温度。

确定该系统在具有辐射和无辐射条件下的时间常数。时间常数 τ 是指瓶子和周围环境的温度差从初始值下降到 63.2% 所用的时间，由下式表示：

$$\frac{T_{amb}-T(\tau)}{T_{amb}-T(0)}=0.368$$

设 $m=3.5$ kg，$c_v=4.2$ J/g K，$A=0.3\ \mathrm{m^2}$，$T(0)=5$℃，$T_{amb}=30$℃。在没有辐射的情况下，设瓶子表面的自然对流传热系数 $h=2\ \mathrm{W/K\ m^2}$。在有辐射的情况下，$\varepsilon=0.5$。

[答案：在无辐射的情况下：$\tau=6.8$ h$(=mc_v/(hA))$；在有辐射的情况下：$\tau=2.78$ h，是初始温度的函数。]

12.3 绝热管子的绝热层外表面通过对流损失的热量由下式决定：

$$q=\frac{2\pi L(T_i-T_\infty)}{\frac{1}{k}\ln(r_o/r_i)+\frac{1}{r_o h}}$$

式中，L 是管子的长度，r_o 是绝热层的外径，r_i 是绝热层的内径，k 是热导，h 是传热系数。当 r_o 取值较小时，附加的绝热层可以提高传热率。为了描述这种效果，将 q

作为r_o的函数绘制出曲线,其中r_o的间隔为$r_o = k/h$。设$h = 5 \text{W/m}^2\text{K}$,$k = 0.1 \text{ W/m K}$,$r_i = 0.01 \text{ m}$,$L = 1 \text{ m}$,$T_i = 100℃$,$T_\infty = 20℃$。一般认为增加绝热层会降低传热率,但这个例子是一个例外。

12.4 温度传感器可以用于测量流动液体的温度。该传感器装在一个小尺寸的圆柱探头上,而该探头沿管壁垂直于流体流向伸入流体中。可以将探头视为肋片,其上的温度分布为:

$$\frac{T(x) - T_\infty}{T_b - T_\infty} = \frac{\cosh m(L - x)}{\cosh mL}$$

其中,

$$m^2 = \frac{hP}{kA_c}$$

T_b是侧壁的温度,T_∞是流体的温度,$P = \pi d$是直径为d的探头周长,$A_c = \pi d^2/4$是横截面面积,k是热导,h是传热系数,L是探头长度。

由于沿着探头有对流发生,所以探测的温度误差e为:

$$e = T(L) - T_\infty = \frac{T_b - T_\infty}{\cosh mL}$$

绘出误差与探头长度的关系曲线,其中$0.005 \leqslant L \leqslant 0.1$ m,k取$20 \leqslant k \leqslant 400$ W/m K中的几个值。设探头的直径为0.005 m,流体的温度为100℃,侧壁的温度为80℃,流体和探头的传热系数为25 W/m^2 K。

12.5 由于存在非线性自然热辐射,所以在辐射路径上隔热板会降低传热效率,因此在需要绝热的情况下,采用隔热板是很重要的。隔热板的作用可以理解为两个无限大平板之间的热传递,这两个平板温度分别为T_1和T_2,且两板间为真空。当表面像黑体一样辐射时,在没有隔热板的情况下传热率q为:

$$q = \sigma(T_1^4 - T_2^4)$$

当有一块隔热板时,传热率由下式决定:

$$q = \sigma(T_1^4 - T_m^4)$$
$$q = \sigma(T_m^4 - T_2^4)$$

其中,T_m是隔热板的温度,当有两个隔热板时,传热率为:

$$q = \sigma(T_1^4 - T_{m1}^4)$$
$$q = \sigma(T_{m1}^4 - T_{m2}^4)$$
$$q = \sigma(T_{m2}^4 - T_2^4)$$

其中,$\sigma = 5.667 \times 10^{-8}$ W/m^2 K^{4}是斯蒂芬-玻耳兹曼常数。T_{m1}和T_{m2}分别为隔热板1和隔热板2的温度。如果这两个平板的温度分别为100℃和20℃,则分别确定在没有隔热板、有一块隔热板和有两块隔热板情况下的传热率。

[答案:没有隔热板时:$q = 680$ W/m^2;有一块隔热板时:$q = 340.1$ W/m^2,$T_m = 340.15$ K;有两块隔热板时:$q = 226.7$ W/m^2,$T_{m1} = 352.2$ K,$T_{m2} = 326.7$ K。]

12.6 普朗克分布表示在某一温度下黑体辐射的单色辐射率,由下式决定:

$$E_{\lambda,b}(\lambda, T) = \frac{C_1}{\lambda^5[\exp(C_2/\lambda T) - 1]}$$

式中，λ 是波长，T 是温度，$E_{\lambda,b}$ 是单色辐射率，$C_1 = 3.742 \times 10^8$ W $\mu m^4/m^2$，$C_2 = 1.439 \times 10^4$ μm K。在辐射计算中，通常需要对该函数在一定波长范围内进行积分。当所有波长进行积分时，得到下式：

$$\int_0^\infty E_{\lambda,b} \mathrm{d}\lambda = \sigma T^4$$

式中，$\sigma = 5.667 \times 10^{-8}$ W/m^2 K^4 是斯蒂芬－玻耳兹曼常数。在 T = 300 K，400 K 和 500 K 时，分别对上式进行数值积分并将结果与精确值比较。注意，数值积分中给出上下限有相当大的困难。对于下限可以用一个非常小的非零数来近似，例如取 0.5 μm。对于上限，其取值主要取决于 T 值，例如可以取 300 μm。

12.7 用 12.3.1 节介绍的方法来确定当 Pr = 0.07，0.7 和 7.0 时温度边界层和速度边界层的厚度。在 12.3.1 节中，边界层温度分布 δ_T 定义为 η 的值，$y_4 = T*(\eta = \delta_T) = 0.99$；边界层速度分布 δ_u 定义为 η 的值，$y_2 = u(\eta = \delta_u) = 0.99$。对于每个普朗特数，用 $P_r^{1/3}$ 的关系比较 δ_u/δ_T 与预测值的差异。

[答案参见表 12.10。]

表 12.10　习题 12.7 的答案

Pr	δ_u	δ_T	δ_u/δ_T	$Pr^{1/3}$
0.07	4.92	13.66	0.36	0.41
0.7	4.92	5.63	0.87	0.89
7.0	4.92	2.45	2.01	1.91

12.8 空气以 u_∞ = 1 m/s 的速度流过平板，Pr = 0.7，$v_{vis} = 1.5 \times 10^{-5}$ m/s^2。用 12.3.1 节给出的公式计算 $T(x, y)$，区间为 $0 < x < x_{crit}$ 和 $0 < y < 10x_{crit}/(\mathrm{Re}_{crit})^{1/2}$，其中，

$$\mathrm{Re}_{crit} = \frac{u_\infty x_{crit}}{v_{vis}} = 5 \times 10^{-5}$$

绘出 $T(x, y)$在这一区域上的等高线图。在 12.3.2 节中 $T(x, y) = y_4$，且：

$$\eta = y\sqrt{u_\infty / v_{vis} x}$$

12.9 与习题 12.8 的条件相同。绘出下面流函数的等高线图：

$$\Psi = f u_\infty \sqrt{\frac{v_{vis} x}{u_\infty}}$$

12.10 利用 12.3.2 节中求解自然对流的方法求 η 的值，条件为 Pr = 0.07，0.7 和 7.0，且速度 u 为最大。在 12.3.2 节中，$u = y_2$。当 u 为负数时，调用函数 fminbnd，根据图 12.13 至图 12.15 来选择区间。

[答案：Pr = 0.7，u_{max} = 0.2784，η_{max} = 0.9532，参见表 12.11。]

12.11 根据 12.3.2 节中求解自然对流的方法，确定在 Pr = 0.07，0.7 和 7.0 时边界层热量和速度分布区间。在 12.3.2 节中，边界层温度分布区间 δ_T 定义为和 η 值相等，$y_4 = T*(\eta = \delta_T) = 0.01$。而且，边界层速度分布区间 δ_u 定义为和 η 值相等，

$y_2 = u(\eta = \delta_u)/u_{max} = 0.01$。其中,$u_{max}$的值是习题 12.4 中的值。

[答案参见表 12.11。]。

12.12 根据 12.3.2 节中所描述的空气流过加热平板上的自然对流,速度分量可以由下式表示:

$$u = \frac{2v_{vis}}{x}Gr_x^{1/2}\frac{df}{d\eta}$$

$$v = \frac{v_{vis}}{x}\left(\frac{Gr_x}{4}\right)^{1/4}\left(\eta\frac{df}{d\eta} - 3f\right)$$

对于静止空气,当温度为 300 K 时计算在 $0 < x < 1$ m, $0 < y < 0.25$ m 范围内的速度分量 $u(x, y)$ 和 $v(x, y)$。在[0.1, 0.5]将 x 分成 5 段,绘出 u 和 v 的曲线图。参考图 12.12 中定义的 x 和 y。

表 12.11 习题 12.10 和习题 12.11 的答案

Pr	0.07	0.7	7.0
η_{max}	1.21	0.96	0.728
u_{max}	0.455	0.278	0.131
δ_u	9.48	5.65	6.41
δ_T	10.27	4.47	1.78

12.13 在 12.3.2 节中定义的流函数如下:

$$\Psi = 4vf\left(\frac{Gr_x}{4}\right)^{1/4}$$

对于温度为 300 K 的气体,绘制出流函数的等高线图,区间为 $0 < x < 1$ m 和 $0 < y < 0.25$ m。参见图 12.12 中 x 和 y 的定义。

12.14 根据 12.4.1 节的分析过程,改变计算角系数的方程形式,使其适用于本例,在本例中矩形位于两个相互垂直的平面。图 12.24 给出了面积微元与有限矩形分布位置的几何图形。为了简化,只考虑面积微元能看到全部有限矩形这一种情况。也就是说,相互垂直的平面的公共交线不能穿过这个有限矩形。根据表 12.12 给出的数据确定图 12.24 所示表面的角系数,注意在这里 $\cos\theta_1 \neq \cos\theta_2$。

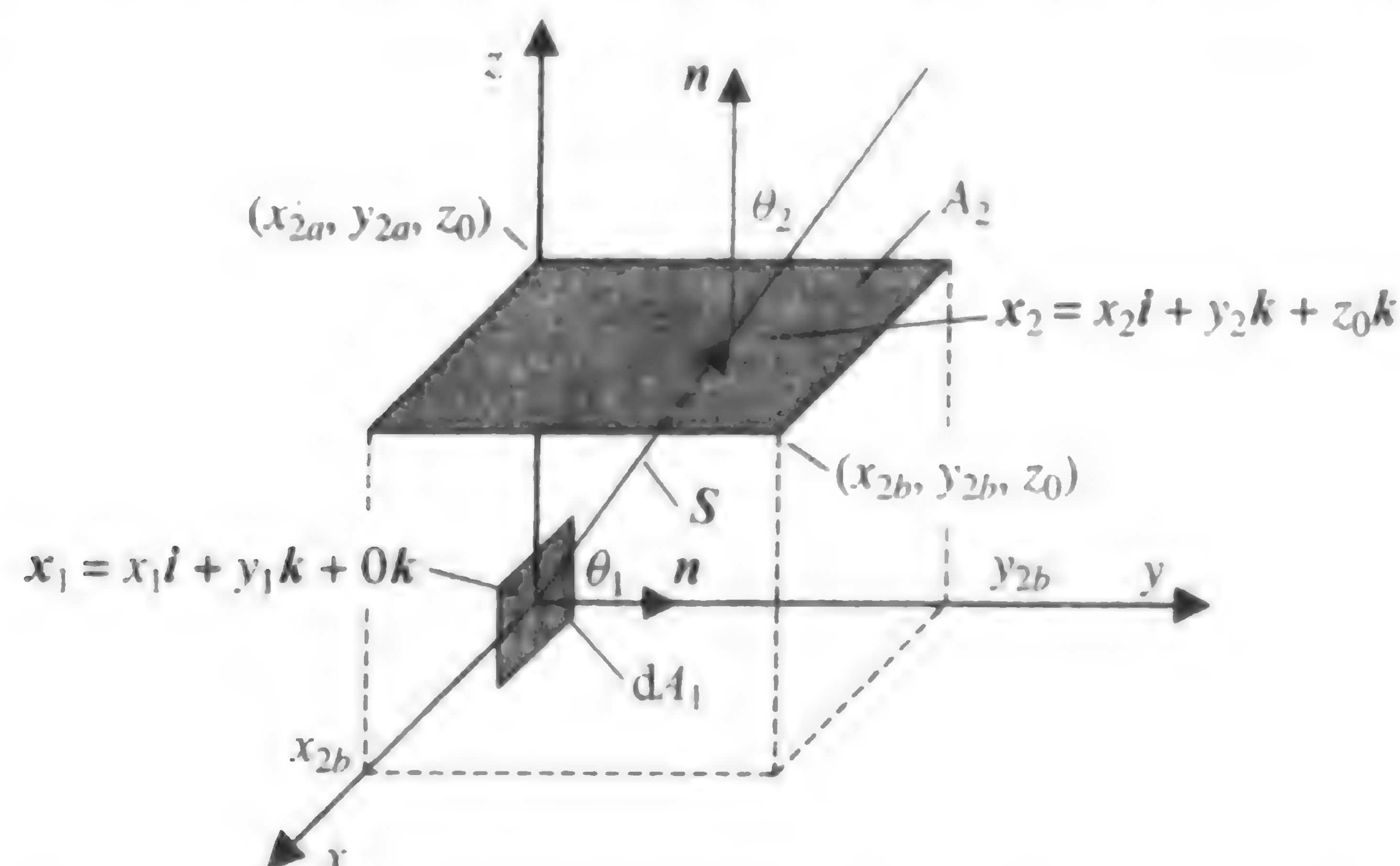


图 12.24 面积微元和有限矩形相互垂直的几何图形

12.15 根据 12.4.3 节对包壳的分析,将实例中表面 #4 分成大小相等的两块后系统有五个面,则应重新计算该例。设表面特性与前例一样,计算每个表面的传热率,并与

先前例子的结果相比较。根据 12.4.3 节给出的表面编号表,表面 4 分成两个部分,上半部分为 # 4,下半部分为 # 5。这种几何分布的角系数计算结果参见表 12.13。这些值是用霍特尔(Hottel)的 crossed-string 方法得到的。结果参见表 12.14。

表 12.12　习题 12.14 的参数和答案

参数	组
有限矩形的几何尺寸	
原点 x 坐标	-1
原点 y 坐标	-1
顶点 x 坐标	0
顶点 y 坐标	0
微元与垂直平面间的距离	5
计算角系数 F_{d_1-2}	0.0012

表 12.13　习题 12.15 的角系数

$i\backslash j$	1	2	3	4	5
1	**0**	**0.3615**	**0.2770**	**0.0957**	0.2658
2	0.2169	**0**	**0.2169**	**0.2831**	0.2831
3	0.2770	0.3615	**0**	**0.2658**	0.0957
4	0.1148	0.5662	0.3190	**0**	0
5	0.3190	0.5662	0.1148	0	0

表 12.14　习题 12.15 的答案

表面 #	1	2	3	4	5
Q(kW)	-8560	8064	4451	-2373	-1582
T(K)	550	700	650	600	600
q(kW/m^2)	-2853	1613	1484	-949	-633

注意:满足能量守恒,即 Q 的和为零。将这些结果和前面章节中的结果相比较,可以看出在这两类计算中,表面 1~3 的传热率(Q)和流量(q)很接近。然而对于表面 4,流量随着长度的变化而变化。有意思的是,表面 4 和 5 总的传热率与原先表面 4 没有分开时的传热率非常接近。

12.16 固体球初始温度均匀分布,边界表面具有对流,球的瞬时温度分布由下式给出①:

$$\theta(\xi,\tau)=\sum_{n=1}^{\infty}C_u\exp(-\zeta_n^2\tau)\frac{\sin(\zeta_n\xi)}{\zeta_n\xi}$$

其中,

$$\theta(\xi,\tau)=\frac{T(\xi,\tau)-T_\infty}{T(\xi,0)-T_\infty}$$

且 $\tau=\alpha t/a^2$,α 是热扩散系数,a 是球体的半径,t 是时间,$\xi=r/a$,r 是球体径向位置,T_∞ 是周围空气的温度,

$$C_n=\frac{4[\sin\zeta_n-\zeta_n\cos\zeta_n]}{2\zeta_n-\sin 2\zeta_n}$$

① 参见 F.P.Incropera 和 D.P.DeWitt,出处同上,第 229 页。

ζ_n 是下列方程的正根:

$$1-\zeta_n\cot\zeta_n=Bi$$

式中,$Bi = ha/k$ 是毕奥特数,h 是传热系数,k 是球体的热导。

当 $0\leqslant\xi\leqslant1$,$0\leqslant\tau\leqslant1.5$ 且 $Bi = 0.5$ 时,绘出 $\theta(\xi,\tau)$ 随 ξ 和 τ 变化的曲线图。

12.17 三角形肋片上的温度如图 12.25 所示,其上的稳态温度分布由下式确定:

$$(1-\eta)\frac{\mathrm{d}^2\theta}{\mathrm{d}\eta^2}-\frac{\mathrm{d}\theta}{\mathrm{d}\eta}-M^2\theta=0$$

式中,$\eta = x/L$,

$$\theta(\eta)=\frac{T(\eta)-T_\infty}{T_\mathrm{b}-T_\infty}$$

$$M^2=\frac{2hL^2}{kt}\sqrt{1+\left(\frac{t}{2L}\right)^2}$$

h 为传热系数,k 为热导,设 $t/w<<1$。

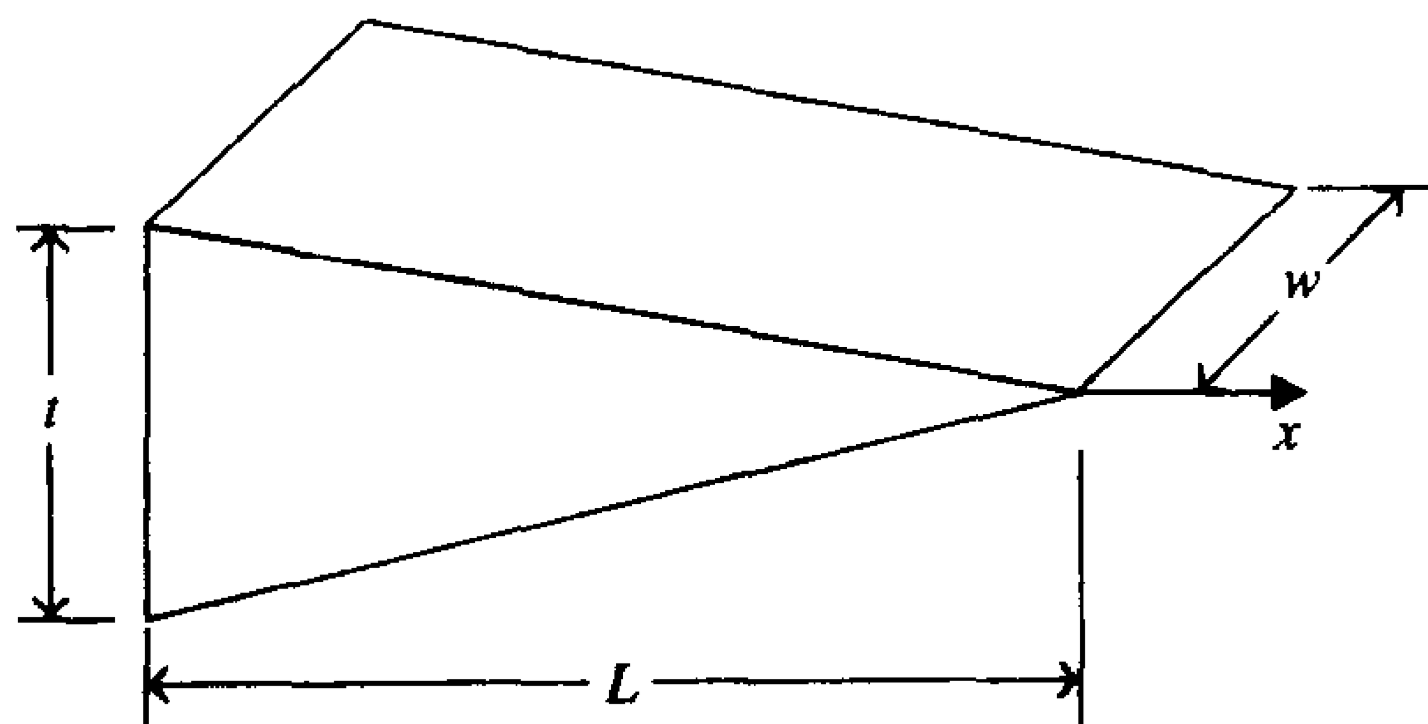


图 12.25 三角形肋片的几何尺寸

设 $\eta = 0$,边界条件 $T(0) = T_\mathrm{b}$,即:

$$\theta(0)=1$$

$\eta = 1$ 时,边界条件为:

$$\left.\frac{\mathrm{d}\theta}{\mathrm{d}\eta}\right|_{\eta=1}=0$$

肋片效率 η_f 由下式获得:

$$\eta_f=\left.\frac{-1}{M^2}\frac{\mathrm{d}\theta}{\mathrm{d}\eta}\right|_{\eta=0}$$

在区间 $0.01<M^2<100$ 内,以 10 的对数坐标等间隔划分,确定肋片效率,并调用函数 semilogx 画出图形。将计算结果与下式的计算结果加以比较[①]:

$$\eta_f=\frac{1}{M}\frac{I_1(2M)}{I_0(2M)}$$

其中,$I_n(x)$为修正的一类 n 阶贝塞尔函数,由函数 besseli 确定。

① 参见 F.P.Incropera 和 D.P.DeWitt,出处同上,第 125 页。

参考文献

1. J.P.Holman, *Heat Transfer*, 7th ed. McGraw-Hill, New York, NY, 1990.
2. F.P.Incropera and D.P.DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed. John Wiley & Sons, New York, NY, 1996.
3. S.Kakac and H.Liu, *Heat Exchangers: Selection, Rating, and Thermal Design*, CRC Press, Boca Raton, FL, 1998.
4. F.Krieth and M.S.Bohn, *Principles of Heat Transfer*, 5th ed. West Publishing Co., New York, NY, 1993.
5. A.F.Mills, *Heat Transfer*, Irwin, Boston, MA, 1992.
6. R.Siegel and J.R.Howell, *Thermal Radiation Heat Transfer*, 3rd ed. Hemisphere Publishing Co., Washington, DC, 1992.
7. N.V.Suryanarayana, *Engineering Heat Transfer*, West Publishing Co., New York, NY, 1995.

第13章 优　　化

本章介绍如何用 MATLAB 的优化工具箱来解决一系列工程优化问题,给出了典型的示例。

13.1 定义、公式及图解

13.1.1 概述

工程优化是在不同约束条件下,求多变量系统最优解的过程。在这一定义中,“最优”一词指在一个或多个设计目标中,决策者希望得到的具有最小或最大性能指标的一种设计目标。例如,希望设计一种可靠性最高、重量最轻与成本最低的产品。在工程优化过程中,需要选择描述系统的变量(例如尺寸、形状、材料、工作性能等)。目标是决策者希望得到的尽可能高或尽可能低的量,约束指对系统的某一技术指标有一定的限制。

一般来说,优化问题是在一定约束条件下对一个或多个目标函数进行最小化,通常用下式描述:

$$\underset{x\in D}{\text{minimize}}\{f_1(x),\ f_2(x)\ ,\cdots,f_m(x)\} \tag{13.1}$$

式中的 $f_i, i=1,\cdots,m$ 是一标量目标函数,该目标函数将向量变量 $\boldsymbol{x}$ 映射到目标空间中。$\boldsymbol{x}$ 为 n 维决策向量,受约束条件限制分布于可行域 D 中。在可行域的说明中,包含上述优化问题的约束条件。一般可行域由 J 个不等式约束条件或 K 个等式约束条件或二者同时构成,如下所示:

$$D = \{x:\ g_j(x) \leqslant 0,\ h_k(x) = 0,\ j = 1,\cdots,\ J,\ k = 1,\cdots,K\} \tag{13.2}$$

当目标函数和约束函数均为变量的线性函数时,称之为线性规划;反之,两者至少有一个为非线性函数时,称之为非线性规划。

表 13.1 中列出了优化问题的分类、对应的 MATLAB 求解函数以及本章提供的示例。

表 13.1　优化问题分类、MATLAB 函数和第 13 章的示例

问题分类	MATLAB 函数	示例
线性规划	linprog	13.2 生产计划
		13.3 炼油问题
非线性规划		
单目标无约束优化		
多变量优化	fminunc	13.1 和 13.4 两弹簧系统
	fminsearch	13.5 瓶底问题
曲线拟合	lsqcurvefit	13.6 应力-应变关系
最小二乘法	lsqnonlin	13.7 应力-应变关系
		13.8 半经验 P-v-T 关系式
		13.9 矿物勘探
单目标约束优化		
单变量优化	fminbnd	13.10 工厂的管道造价问题
		13.11 封闭盒

（续表）

问题分类	MATLAB 函数	示例
多变量优化	fmincon	13.12 两杆架构问题
		13.13 螺旋形受压弹簧
		13.14 齿轮减速箱
二次项优化	quadprog	13.15 生产计划
半无穷优化	fseminf	13.16 平面两连杆操纵器
多目标优化	fminimax	13.17 振动台问题
	fgoalattain	13.18 生产计划

13.1.2 图解法

两变量优化问题的求解结果可由 MATLAB 的画点功能以图形方式显示。下面给出一个例子。

例 13.1　两弹簧系统

图 13.1 中是两个弹簧组成的系统[①]。图 13.1 中的虚折线是施加载荷前弹簧未变形的情况，图13.1中的实折线是施加一载荷后弹簧变形的情况。在 A 点施加一载荷后系统处于变形状态，经过一定时间后将停留在平衡点 B，如实线所示。要研究的是系统在受力情况下的平衡状态，即 B 点的位置(x_1, x_2)。

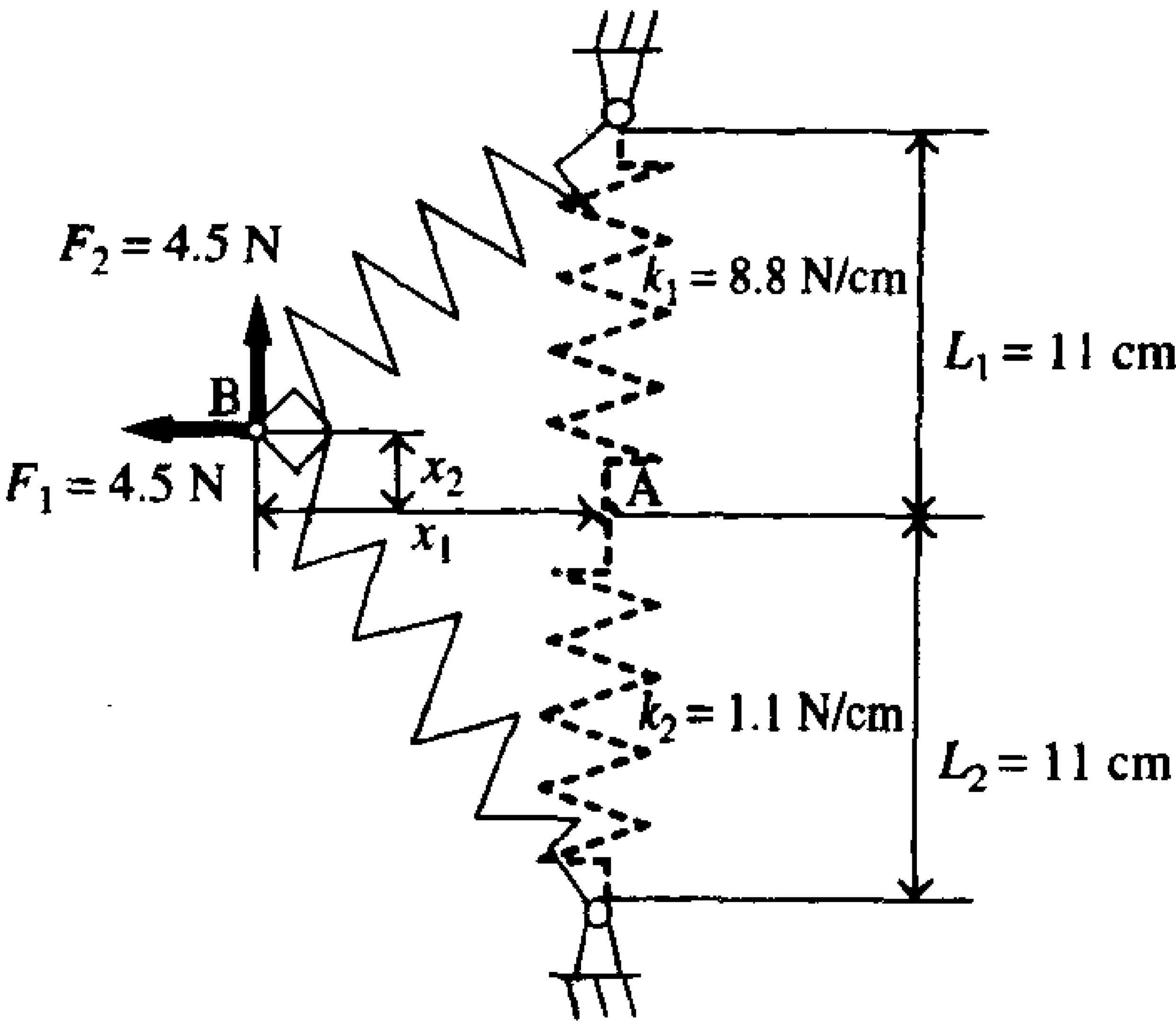


图 13.1　两弹簧系统

为得到系统的平衡状态，首先要算出系统的势能(PE, potential energy)，然后相对于设计变量 x_1 和 x_2 值求最优值从而求得 B 点位移。势能由弹簧伸长时弹性势能与弹簧在平稳位置处的弹性势能之差计算得出，如式(13.3)中的前两项所示；外力所做的功在式(13.3)中由后两项给出。式中的参数 k_1, k_2, L_1, L_2, F_1 和 F_2 均为常量，其数值如图 13.1 所示。至此，无约束优化问题的目标函数可以得出：

$$\underset{x_1, x_2}{\text{minimize}} \text{PE}(x_1, x_2) = 0.5k_1\left(\sqrt{x_1^2 + (L_1 - x_2)^2} - L_1\right)^2 + 0.5k_2\left(\sqrt{x_1^2 + (L_2 + x_2)^2} - L_2\right)^2 - F_1x_1 - F_2x_2 \tag{13.3}$$

① G. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, NY, 1984, 第 72 ~ 73 页。

目标函数中有两个变量 x_1,x_2,其近似值可由图形估计出。程序如下:

```
k1 = 8.8;k2 = 1.1;L1 = 11;L2 = 11;F1 = 4.5;F2 = 4.5;
[x1,x2] = meshgrid(linspace( - 5,15,15),linspace( - 5,15,15));
PE1 = 1/2 * k1 * (sqrt(x1.^2 + (L1 - x2).^2) - L1).^2;
PE2 = 1/2 * k2 * (sqrt(x1.^2 + (L2 + x2).^2) - L2).^2;
PE = PE1 + PE2 - F1 * x1 - F2 * x2;
subplot(1,2,1);
h = contour(x1,x2,PE,[ - 40:20:20,50:100:500],'k');
clabel(h);
axis([ - 5,15, - 5,15])
xlabel('x _ 1');
ylabel('x _ 2');
subplot(1,2,2);
surfc(x1,x2,PE);
axis([ - 10,15, - 10,15, - 100,500]);
zlabel('PE');
xlabel('x _ 1');
YLABEL('x _ 2')
```

程序执行结果如图 13.2 所示。图 13.2(a)是 PE 的等高线图,图上标出了每条等高线的数值,从图中可清楚地找到极小值/极大值的分布。图 13.2(b)中上面是 PE 的三维图形,三维图形下面是等高线图,图中也标出了极小值/极大值所在点的近似位置,在该点 PE 目标函数取得极小值/极大值。该例中极小值/极大值所在点的精确位置将在 13.3.1 节中给出,在 13.3.1 节中目标函数的优化是由这个无约束优化函数 fminunc 实现的。

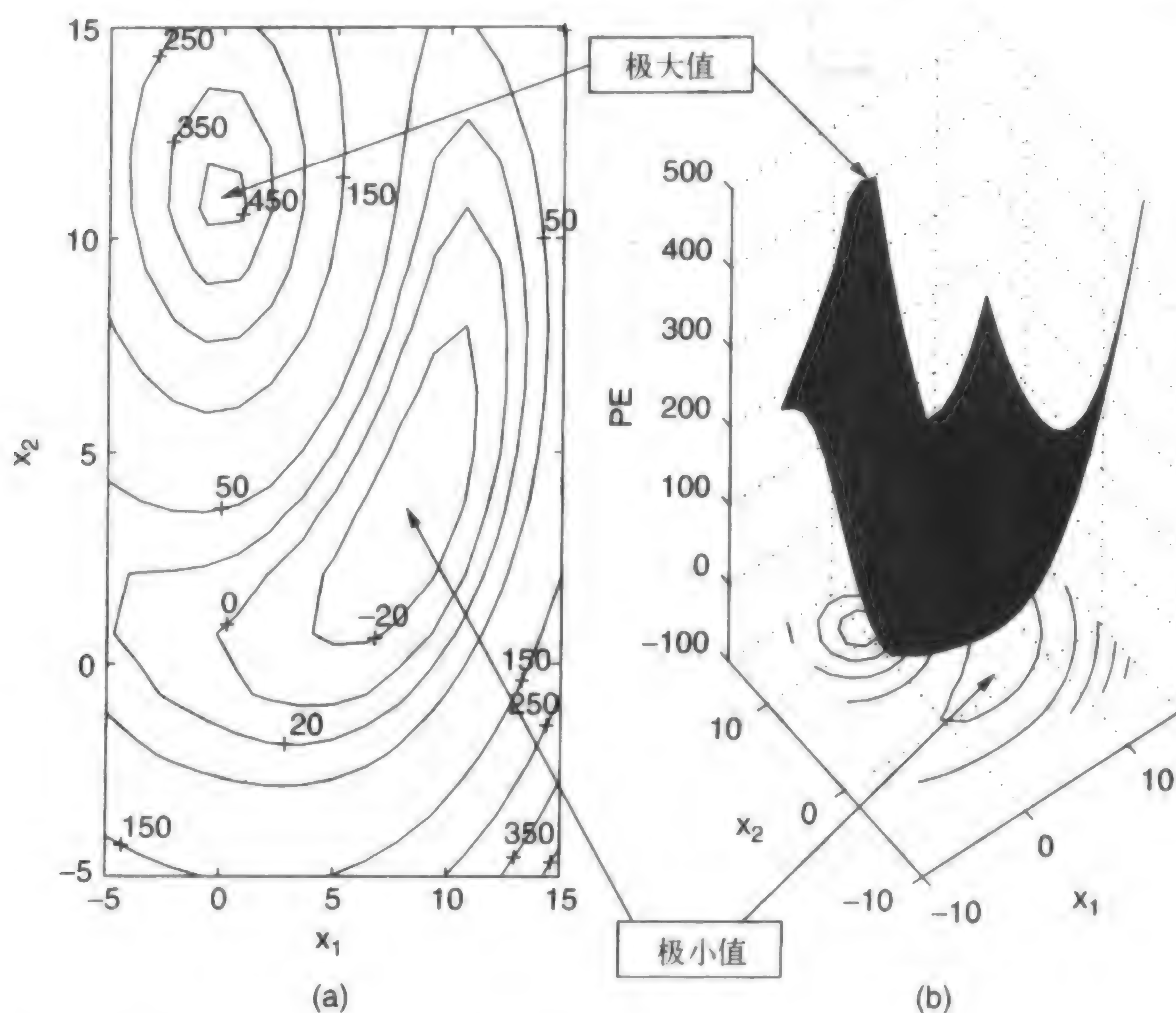


图 13.2　针对图 13.1 给出的两弹簧系统的 PE 函数图:(a) 等高线图;(b) 曲面图

13.2　线性规划

线性规划(LP, linear programming)是一种优化方法，在优化问题中目标函数和约束函数均为设计变量的线性函数，LP 问题可描述为：

$$\begin{aligned} &\underset{x}{\text{minimize}}\, \boldsymbol{f}^T\boldsymbol{x} \\ &\text{s.t.}: \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b} \\ &\quad \boldsymbol{A}_{eq}\boldsymbol{x} = \boldsymbol{b}_{eq} \\ &\quad \boldsymbol{lb} \leqslant \boldsymbol{x} \leqslant \boldsymbol{ub} \end{aligned} \tag{13.4}$$

其中，$\boldsymbol{f}$, $\boldsymbol{b}$, $\boldsymbol{b}_{eq}$, $\boldsymbol{lb}$, $\boldsymbol{ub}$ 为向量，$\boldsymbol{A}$ 和 $\boldsymbol{A}_{eq}$ 为矩阵，$\boldsymbol{x}$ 为一设计向量变量，上标 T 表示转置，矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 是线性不等式约束条件的系数，$\boldsymbol{A}_{eq}$ 和 $\boldsymbol{b}_{eq}$ 是等式约束条件的系数。在 MATLAB 中，用于 LP 的求解函数为 `linprog`，式(13.4)描述的问题可由 `linprog` 实现求解。`linprog` 函数的调用格式为：

```
[xopt,fopt] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
```

该函数返回一设计向量变量 $xopt = [x_{1opt}, x_{2opt}, \cdots]$ 和标量 $f(x_{opt})$，$x0$ 为 x 的起始点，*options* 为 `optimset` 函数中定义的参数的值。如没有等式约条件，则令 $A_{eq} = [\,]$，$b_{eq} = [\,]$。

下例为 `linprog` 函数的用法。

例 13.2　生产计划

有两种液体产品 A 和 B，需要在两个车间加工处理。每件产品 A 在第一车间的处理时间为 1 小时，在第二车间的处理时间为 1.25 小时；每件产品 B 在第一车间的处理时间为 1 小时，在第二车间的处理时间为 0.75 小时。每个车间每月有 200 小时的时间可以利用，而且 B 产品的市场需求量最大为 150 件。假定 A 产品和 B 产品的利润每件分别为$ 4 和 $ 5，现在需要求出使生产商的利润最大时 A 产品和 B 产品的生产量。

首先，假定 x_1 为 A 产品的生产数量，x_2 为 B 产品的生产数量。则目标函数和约束条件为[①]：

$$\begin{aligned} &\text{minimize} f(x_1, x_2) = -4x_1 - 5x_2 \\ &\text{s.t.}: \\ &\quad g_1: x_1 + x_2 \leqslant 200 \\ &\quad g_2: 1.25x_1 + 0.75x_2 \leqslant 200 \\ &\quad g_3: x_2 \leqslant 150 \\ &\quad\quad (x_1, x_2) \geqslant 0 \end{aligned} \tag{13.5}$$

所以，

$$f^T = [-4\ -5] \tag{13.6}$$

不等式约束表示为：

① 参见 A. Osyczka, *Multicriterion Optimization in Engineering with Fortran Programs*, Ellis Horwood Limited, West Sussex, England, 1984 年，第 4 页。

$$\begin{bmatrix} 1 & 1 \\ 1.25 & 0.75 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 200 \\ 200 \\ 150 \end{bmatrix} \tag{13.7}$$

程序为：

```
f=[-4,-5];
A=[1,1;1.25,0.75;0,1];
b=[200,200,150];
lb=[0,0];
x=linprog(f,A,b,[],[],lb,[])
```

程序的执行结果是A产品的生产数量为 $x_1=50$，B产品的生产数量为 $x_2=150$。

例13.3　炼油问题

炼油厂炼油需要三种类型的原油：C1型，C2型和C3型。C1型原油成本为$0.40/加仑，从市场每天最多可获得10 000加仑；C2型原油成本为$0.20/加仑，每天最多可获得12 000加仑；C3型原油成本为$0.10/加仑，每天最多可获得15 000加仑。炼油厂将每种类型的原油加工成汽油，生产的汽油有三种类型：常规汽油、附加油和高级汽油。三种类型汽油的市场最大需求量分别为常规汽油9000加仑/天，附加油8000加仑/天，高级汽油7000加仑/天。炼油厂卖给发售商的汽油价格分别为：常规汽油$0.70/加仑，附加油$0.80/加仑，高级汽油$0.90/加仑。假定1加仑C1型原油能生产0.2加仑的常规汽油，0.3加仑的附加油，0.5加仑的高级汽油；1加仑C2型原油能生产0.5加仑的常规汽油，0.3加仑的附加油，0.2加仑的高级汽油；1加仑C3型原油能生产0.7加仑的常规汽油，0.3加仑的附加油，不能生产高级汽油。求炼油厂需要分别购买原油C1，C2和C3各多少加仑时才能使其日利润最大。

设 x_1、x_2 和 x_3 分别表示要购买的C1型原油、C2型原油和C3型原油的加仑数。下面分别是目标函数 f、原油的约束条件及可生产的汽油约束条件：

$$\begin{aligned} \text{minimize } f(x_1,x_2,x_3) &= -[0.7(0.2x_1+0.5x_2+0.7x_3) \\ &\quad +0.8(0.3x_1+0.3x_2+0.3x_3) \\ &\quad +0.9(0.5x_1+0.2x_2)-0.4x_1-0.2x_2-0.1x_3] \\ &= -0.43x_1-0.57x_2-0.63x_3 \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & g_1: 0.2x_1+0.5x_2+0.7x_3 \leqslant 9000 \\ & g_2: 0.3x_1+0.3x_2+0.3x_3 \leqslant 8000 \\ & g_3: 0.5x_1+0.2x_2 \leqslant 7000 \\ & g_3: x_1 \leqslant 10\,000 \\ & g_3: x_2 \leqslant 12\,000 \\ & g_3: x_3 \leqslant 15\,000 \\ & (x_1,x_2,x_3) \geqslant 0 \end{aligned} \tag{13.8}$$

程序为：

```
f=[-0.43,-0.57,-0.62];
A=[0.2,0.5,0.7;0.3,0.3,0.3;0.5,0.2,0;1,0,0;0,1,0;0,0,1];
```

```
b=[9000,8000,7000,10000,12000,15000];
lb=[0,0,0];
[x,f]=linprog(f,A,b,[],[],lb,[])
```

执行后得到 C1 型原油、C2 型原油、C3 型原油的最优加仑数分别是 x_1 = 9200, x_2 = 12 000, x_3 = 1657, 日利润为$ 11 840。

13.3 非线性规划

非线性规划(NLP,nonlinear programming)指目标函数或约束函数(或两者)为设计变量的非线性函数的一种优化方法。NLP 问题及其相关问题分为两类:无约束优化方法和约束优化方法。

13.3.1 无约束优化

无约束 NLP 方法是确定下列函数式(13.5)表示的无约束变量函数极小值的一种方法:

$$\underset{x}{\text{minimize}} f(x) \tag{13.9}$$

这里,x 为设计向量变量,f 为一标量目标函数。解决上述问题可由两个函数实现,即 fminunc 函数和 fminsearch 函数。其中,fminunc 函数基于梯度搜索法实现,而 fminsearch 函数基于直接搜索法实现。调用 fminunc 的命令如下:

```
[xopt,fopt]=fminunc(@UserFunction,x0,options,p1,p2,...)
```

其中,**UserFunction** 是求解目标函数 f 的函数文件名,参数 $x0$ 表示 x 的起始值,是向量,*options* 为 optimset 定义的参数赋值。$p1,p2$ 等为传递给 **UserFunction** 的参数。

调用 fminsearch 的命令如下:

```
x=fminsearch(@UserFunction,x0,options,p1,p2,...)
```

其中,各个参数的定义与 fminunc 一致。

下面介绍 fminunc 和 fminsearch 的用法。

例 13.4 两弹簧系统

现在,将例 13.1 的两弹簧系统作为无约束优化问题进行数值分析。首先,创建设计变量 x_1 和 x_2 的无约束目标函数 **SpringEquilibrium**,程序如下:

```
function Twospring
x0=[0.5,5];
k1=8.8;k2=1.1;L1=11;L2=11;F1=4.5;F2=4.5;
options=optimset('LargeScale','off');
[x,f]=fminunc(@SpringEquilibrium,x0,options,k1,k2,L1,L2,F1,F2)

function PE=SpringEquilibrium(x,k1,k2,L1,L2,F1,F2)
PE1=1/2*k1*(sqrt(x(1)^2+(L1-x(2))^2)-L1)^2;
PE2=1/2*k2*(sqrt(x(1)^2+(L2+x(2))^2)-L2)^2;
PE=PE1+PE2-F1*x(1)-F2*x(2);
```

结果为 $x=[8.4251,3.6331]$,$f=$ PE = −35.0507。与图 13.2 中的近似解一致。要注意的是,fminunc 默认情况下选择大规模优化算法,而将 *options* 参数的'LargeScale'设为'off'时,使用中规模优化算法。要获得 PE 函数的极大值,可通过求(1/PE)或(−PE)函数的最优解得到。这时,创建 **SpringEquilibriumMax** 函数,程序为:

```
function Twospring2
x0 = [0.5,5];
k1 = 8.8;k2 = 1.1;L1 = 11;L2 = 11;F1 = 4.5;F2 = 4.5;
options = optimset('LargeScale','off');
[x,f] = fminunc(@SpringEquilibriumMax,x0,options,k1,k2,L1,L2,F1,F2)

function PE = SpringEquilibriumMax(x,k1,k2,L1,L2,F1,F2)
PE1 = 1/2 * k1 * (sqrt(x(1)^2 + (L1 - x(2))^2) - L1)^2;
PE2 = 1/2 * k2 * (sqrt(x(1)^2 + (L2 + x(2))^2) - L2)^2;
PE = -(PE1 + PE2 - F1 * x(1) - F2 * x(2));
```

改变后的执行结果为 $x=[0\ 11]$,PE = 549.45。与图 13.2 得到的近似值相比,向量 x 的值是一致的。

需要指出的是,fminunc 和 fminsearch 及本章其他所有函数(除 linprog 之外)得到的仅是局部优化解,且解的大小取决于初始点 x0 的位置。有些情况下根据实际问题,如果初始点位置改变,则所求解也会改变。下面的例子将说明这种情况。

例 13.5　瓶底问题

考虑以下二变量函数[①]:

$$\text{minimize } f(x_1,x_2) = (C+1)/8$$

其中:

$$\begin{aligned} C &= \frac{1}{2}(A^2+B^2+3)+\sin(A^2+B^2+2) \\ A &= 6x_1-3 \\ B &= 6x_2-3 \\ 0 &\leqslant (x_1,x_2) \leqslant 1 \end{aligned} \tag{13.10}$$

首先编写以下程序,产生函数的表面图,如图 13.3 所示:

```
[x1,x2] = meshgrid(linspace(0,1,50),linspace(0,1,50));
A = 6 * x1 - 3;B = 6 * x2 - 3;
C = (A.^2 + B.^2 + 3)./2 + sin(A.^2 + B.^2 + 2);
bottle = (C + 1)/8;
colormap([1 1 1]);
surf(x1,x2,bottle);
zlabel('f');
xlabel('x_1');
ylabel('x_2');
view(-10,45)
```

如图 13.3 所示,函数表面图是关于轴对称的,在中央有一圆盖形区域类似瓶子的底部。

① 修订自:D. A. Van Veldhuizen 和 G. B. Lamont,"Multi-Objective Evolutionary Algorithm Research: A History and Analysis," Technical Report TR-98-03, Air Force Institute of Technology, Wright Patterson AFB, OH, 1998。

图 13.3 中有三个区域，每个区域沿一个圆圈有无限个极大值。这三个区域分别为：位于中央的圆盖型局部区域、中间的极大值区域及位于两者之上，处于一个圆的 4 个角的区域。下面是用函数 fminsearch 求上述函数的极大值的程序：

```
x0 = [0.2,0.8];
bottle = inline('-(((6*x(1)-3)^2+(6*x(2)-3)^2+3)/2+sinc(6*x(1)-3)^2…(6*x(2)-3)
      ^2+2)+1)/8','x');
options = optimset('Large','off');
[x,f] = fminsearch(bottle,x0,options)
```

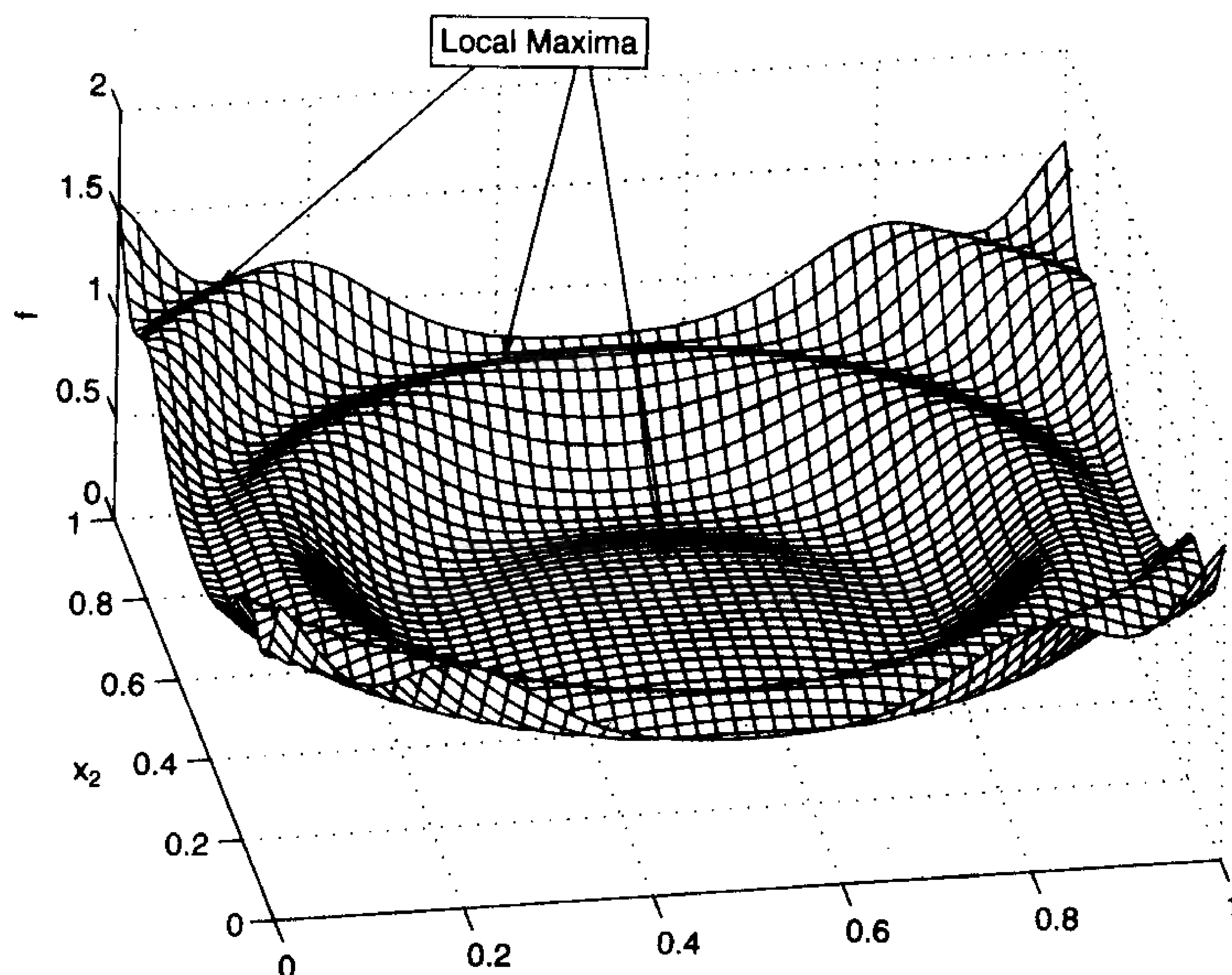


图 13.3 瓶底函数的表面图

执行时发现极大值产生在 x = [0.2068, 0.802]处，且 f = 0.8194。然而，如果将初始点改为 $x0$ = [0.2, 0.9]，则会发现极大值产生在 x = [0.1803, 0.9995]处，且 f = 1.2121；如果将初始点改为 $x0$ = [0.3, 0.6]，则会发现极大值产生在 x = [0.4526, 0.4806]处，且 f = 0.4267。实际上，该函数有无限个极大值，由 fminsearch 得到的局部最优值由初始点决定。

13.3.2 曲线拟合

非线性曲线拟合问题可由求解非线性最小二乘问题的函数 lsqcurvefit 实现。给定输入变量 *xdata*，及其对应的输出变量 *ydata*，由 lsqcurvefit 函数得到式 $f(x, xdata_i)$最接近的系数 x，即：

$$\underset{x}{\text{minimize}} \frac{1}{2} \sum_{i} [f(x, xdata_i) - ydata_i]^2 \tag{13.11}$$

将 *options* 的'LargeScale'设为'off'，MATLAB 用两种优化方法求解式(13.11)描述的问题，函数 lsqcurvefit 的调用格式为：

```
[xopt,resnorm] = lsqcurvefit(@UserFunction,x0,xdata,ydata,lb,ub,options,p1,p2,…)
```

其中，$xopt = x_{opt}$ 为 x 的最优解，*resnorm* 是参差，其欧几里得(Euclidean)范数如下。

$$\sum (f(x_{data}) - y_{data})^2$$

UserFunction 为求解目标函数的文件名,也可以用 inline 创建需要多个输入参数的函数,只须在 inline 语句中定义表达式的同时列出输入参数的名称,如下例所示。参数 *x0* 为起始值,*xdata* 和 *ydata* 分别为输入/输出向量,*options* 为由optimset定义的参数赋值。参数 *lb* 和 *ub* 为向量,分别表示 *x* 的下界和上界,即:$lb \leqslant x \leqslant ub$。

如果不需要 *lb* 和 *ub* 参数,其后跟空矩阵符号[]。*options* 参数后跟空矩阵符号[]时,表示用其默认值。*p*1,*p*2 等为传递给 **UserFunction** 的参数。

下例给出了 lsqcurvefit 函数的用法。

例 13.6 应力-应变关系

表 13.2 列出了一种塑料材料的一组应力-应变数据,表中 σ 为应力值,单位为 ksi。ε 为应变值。假定其关系式具有以下形式:

$$\varepsilon = a + b \ln \sigma \tag{13.12}$$

表 13.2 塑料材料的应力-应变数据

σ	ε
925	0.11
1125	0.16
1620	0.35
2125	0.48
2625	0.61
3125	0.71
3625	0.85

现在要求设计变量 *a* 和 *b*,使上式成为表 13.2 中数据的最适合函数。下面的程序中用内嵌函数定义 **SigmaEpsilonFit** 函数,然后在 lsqcurvefit 函数的第一个参数中引用 **SigmaEpsilonFit** 函数。程序如下所示:

```
sigma = [925,1125,1625,2125,2625,3125,3625];
epsilon = [0.11,0.16,0.35,0.48,0.61,0.71,0.85];
x0 = [0.1,0.1];
SigmaEpsilonFit = inline('x(1) + x(2) * log(sigma)','x','sigma');
[x,resnorm] = lsqcurvefit(SigmaEpsilonFit,x0,sigma,epsilon)
```

执行结果为:$x(1) = a = -3.581$,$x(2) = b = 0.5344$,残差为 *resnorm* = 0.0064,得到最合适的函数为:

$$\varepsilon = -3.581 + 0.5344 \ln \sigma \tag{13.13}$$

13.3.3 最小二乘法

lsqnonlin 函数可对非线性最小二乘问题求解,此外也可用于曲线拟合优化中,如例 13.7 所示。但 lsqnonlin 主要用于具有多组输入数据和一组输出数据的问题,如例 13.8 所示。lsqnonlin 函数求出取得极值处的 *x* 值,问题描述如下:

$$\underset{x}{\text{minimize}} \sum_i [f_i(x)]^2 \tag{13.14}$$

lsqnonlin 函数的调用方法如下：

```
[xopt,resnorm] = lsqnonlin(@UserFunction,x0,lb,ub,options,p1,p2,…)
```

其中，

$$resnorm = \sum_i f_i^2(x_{opt})$$

UserFunction 为求解目标函数的函数文件名，**UserFunction** 创建的目标函数为 $f_i(x)$，而不是 $f_i^2(x)$。注意，不应该显式地构建平方和（参考下例）

$$\sum_i f_i^2$$

参数 *x*0 为起点向量，*lb*，*ub* 为 *x* 的下界和上界，*options* 为 optimset 定义的参数赋值，*p*1，*p*2 等为传递给 **UserFunction** 的参数。如果 *lb* 和 *ub* 之后用到空矩阵符[]，则表示它们未定义，*options* 后有空矩阵符时表示用默认值。

下面介绍 lsqnonlin 函数的用法。

例 13.7　应力-应变关系

现在用 lsqnonlin 函数求解例 13.6 中的应力-应变问题。首先，通过求下式的极小值来确定设计变量 *a* 和 *b*：

$$\underset{a,b}{\text{minimize}} \sum_{i=1}^{7} [\varepsilon_i - (a + b\ln\sigma_i)]^2 \tag{13.15}$$

其中，ε_i 和 σ_i 为表 13.2 列出的实验数据。

程序如下：

```
sigma = [925,1125,1625,2125,2625,3125,3625];
epsilon = [0.11,0.16,0.35,0.48,0.61,0.71,0.85];
x0 = [0.1,0.1];
SigmaEpsilonLeastSq = inline('epsilon - (x(1) + x(2) * log(sigma))','x','sigma',…'epsilon');
[x,resnorm] = lsqnonlin(SigmaEpsilonLeastSq,x0,[],[],[],sigma,epsilon)
```

执行结果为 $x(1) = a = -3.581$，$x(2) = b = 0.5344$，*resnorm* = 0.0064，与例 13.6 得到的结果相同。

例 13.8　半经验的 *P*-*v*-*T* 关系

众所周知，真实气体的 *P*-*v*-*T* 关系式偏离于理想气体的 *P*-*v*-*T* 关系。理想气体的 *P*-*v*-*T* 关系式为：

$$Pv = RT \tag{13.16}$$

式中，*P* 是气体的压强，单位为 atm；*v* 为克分子体积，单位为 cm^3/g mol；*T* 为热力学温度，单位为 K；*R* 为气体常数，*R* = 82.06 atm cm^3/g mol K。在理想气体关系式基础上，修正后的半经验关系式为①：

$$P = \frac{RT}{v-b} - \frac{a}{v(v+b)\sqrt{T}} \tag{13.17}$$

① 参见 G. V. Reklaitis 等，*Engineering Optimization*，John Wiley & Sons，New York，NY，1983 年，第 20～22 页。

其中,a,b 的值由实验数据得出。表 13.3 中列出了气体 P-v-T 参数的一些实验数据。通过求以下最小二乘目标函数的极小值可得出设计向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的值:

$$\underset{a,b}{\text{minimize}} \sum_{i=1}^{8} \left[P_i - \frac{RT_i}{v_i - b} + \frac{a}{v_i(v_i + b)\sqrt{T_i}} \right]^2 \tag{13.18}$$

式中,P_i,v_i 和 T_i 为表 13.3 中第 i 次的实验数据。如果式(13.17)描述的半经验关系式非常接近实验数据,则在最优点处式(13.18)的目标函数应正好等于 0。但是,由于实验误差及方程的简单化而不能精确地表达非线性,因而在最优点处实际上不会为 0。

程序为:

```
x0 = [8000,40];R = 82.06;
T = [283,313,375,283,313,375,283,375];
v = [480,480,576,672,576,672,384,384];
P = [32.7,42.6,44.5,25.7,36.6,38.6,37.6,63.0];
pvt = inline('P - R * T./(v - x(2)) + x(1)./(sqrt(T). * v. * (v + x(2)))','x','R','T','v','P');
format long e;
options = optimset('MaxFunEvals',600);
[x,resnorm] = lsqnonlin(pvt,x0,[],[],options,R,T,v,P)
format short
```

执行结果为 $a = x(1) = 7.422 \times 10^7$,$b = x(2) = 30.682$ 和 $resnorm = 13.8$,由以上值可得出最适合的函数为:

$$P = \frac{RT}{v - 30.682} - \frac{7.442 \times 10^7}{v(v + 30.682)\sqrt{T}} \tag{13.19}$$

表 13.3 气体的 P-v-T 数据

Run i	P(atm)	v,(cm³/g mol)	T(K)
1	32.7	480	283
2	42.6	480	313
3	44.5	576	375
4	25.7	672	283
5	36.6	576	313
6	38.6	672	375
7	37.6	384	283
8	63.0	384	375

该例中所求解对初始点 $x0$ 很敏感。$x0$ 点的值影响 $resnorm$ 的值,要找的是使 $resnorm$ 最小的解。理想情况下使 $resnorm = 0$ 的解是最优解。

例 13.9 矿物勘探

一个 2 km×2 km 的区域被认为覆盖了一层很厚的矿物层。要建立矿物层剖面的模型及开采的经济可行性模型,首先进行初级表面勘探。钻 16 个钻孔,根据记录的矿物层的上下限对每个钻孔钻约 45 m 深。表 13.4 列出了钻孔数据,其中的钻孔坐标与深度以米为

单位[①]。

收集钻孔数据，下一步要建立一个表面矿物层的简单三维计算模型。由 2 km×2 km 的位置边界确定四条垂直边。顶部和底部的平面由式(13.20)定义：

$$z(x,y) = a_0 + a_1 x + a_2 y \tag{13.20}$$

下面的程序使用函数 lsqnonlin 求最优值，并根据表 13.4 中的数据创建顶部和底部平面：

```
xb = [10.0,750.0,1250.0,1990.0,10.0,750.0,1250.0,1990.0,…
    10.0,750.0,1250.0,1990.0,10.0,750.0,1250.0,1990.0];
yb = [10.0,10.0,10.0,10.0,750.0,750.0,750.0,750.0,1250.0,…
    1250.0,1250.0,1250.0,1990.0,1990.0,1990.0,1990.0];
zbtop = [-30.5,-29.0,-28.0,-26.6,-34.2,-32.8,-31.8,-30.3,…
    -36.7,-35.2,-34.2,-32.8,-40.4,-39.0,-38.0,-36.5];
abbot = [-40.5,-39.8,-39.3,-38.5,-41.4,-40.6,-40.1,-39.4,…
    -42.0,-41.2,-40.7,-40.0,-42.8,-42.1,-41.6,-40.9];
x0 = [1,1,1];
MinrlDepErrTop = inline('zbtop - (x(1) + x(2) * xb + x(3) * yb)','x','xb','yb','zbtop');
[xtop,Errornormtop] = lsqnonlin(MinrlDepErrTop,x0,[],[],[],xb,yb,zbtop)
MinrlDepErrBot = inline('zbbot - (x(1) + x(2) * xb + x(3) * yb)','x','xb','yb','zbbot');
[xbot,Errornormbot] = lsqnonlin(MinrlDepErrBot,x0,[],[],[],xb,yb,zbbot)
[xb,yb] = meshgrid(linspace(0,2000,20),linspace(0,2000,20));
zb = xtop(1) + xtopt(2) * xb + xtop(3) * yb;
mesh(xb,yb,zb);
hold on
zb = xbot(1) + xbot(2) * xb + xbot(3) * 3yb;
mesh(xb,yb,zb);
xlabel('x');
ylabel('y');
zlabel('z');
```

表 13.4　表面矿物勘探数据

钻孔数	坐标[x,y]	深度(z)[顶部,底部]
1	[10,10]	[-30.5,-40.5]
2	[750,10]	[-29,-39.8]
3	[1250,10]	[-28,-39.3]
4	[1990,10]	[-26.6,-38.5]
5	[10,750]	[-34.2,-41.4]
6	[750,750]	[-32.8,-40.6]
7	[1250,750]	[-31.8,-40.1]
8	[1990,750]	[-30.3,-39.4]
9	[10,1250]	[-36.7,-42]
10	[750,1250]	[-35.2,-41.2]

① 参见 M. Austin 和 D. Chancogne, *Engineering Programming in C, MATLAB, and JAVA*, John wiley & Sons, New York, NY, 1998, 第 461 页。

(续表)

钻孔数	坐标[x,y]	深度(z[顶部,底部])
11	[1250,1250]	[-34.2,-40.7]
12	[1990,1250]	[-32.8,-40]
13	[10,1990]	[-40.4,-42.8]
14	[750,1990]	[-39,-42.1]
15	[1250,1990]	[-38,-41.6]
16	[1990,1990]	[-36.5,-40.9]

执行程序后得到两个最优平面，如图 13.4 所示。顶部和底部平面的拟和误差均为 0.0122。

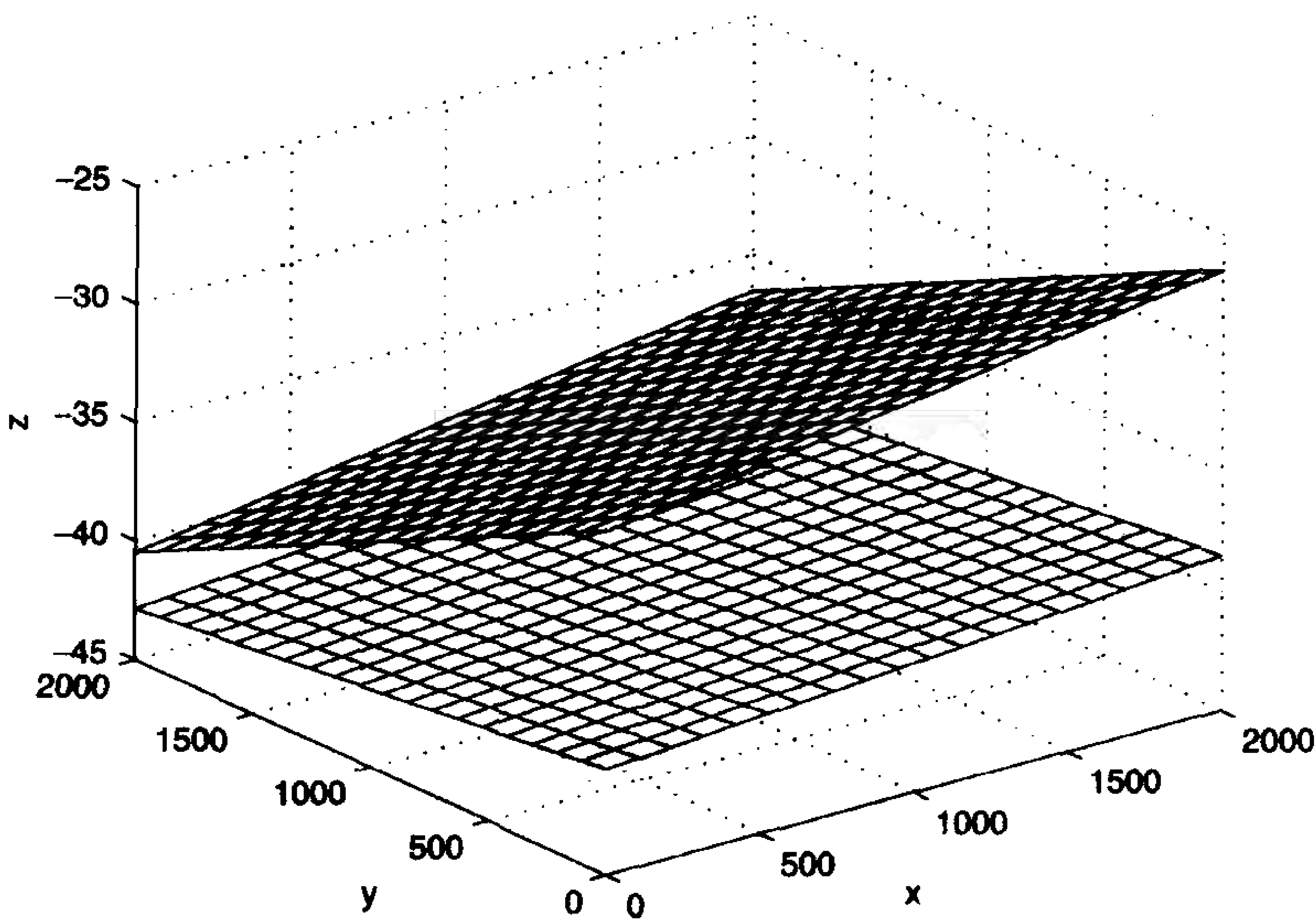


图 13.4 例 13.9 中顶部和底部拟合曲线

13.4 单目标约束优化

非线性约束优化是对类似式(13.1)和式(13.2)描述的受约束目标函数求最优解的一种方法，当 $m=1$ 时称为单目标函数

13.4.1 单变量约束优化

单变量约束优化是对分布于固定区间上的单变量目标函数求最优解的一种方法

$$\underset{x}{\text{minimize}}\, f(x)$$
$$\text{s.t.}: x_1 \leqslant x \leqslant x_2 \tag{13.21}$$

MATLAB 命令为：

```
[xopt,fxopt] = fminbnd(@UserFunction,x1,x2,options,p1,p2,…)
```

式中，$xopt = x_{opt}$ 为 x 的最优值，$fxopt = f(x_{opt})$，**UserFunction** 为求解目标函数的函数文件名。参数 $x1, x2$ 为 x 所属的区间范围，*options* 为由 optimset 中定义的参数赋值，$p1, p2$ 等为传递给 **UserFunction** 的参数。下例给出了 fminbnd 函数的用法。

例 13.10 工厂的管道费用问题

若设计一化工厂，管道费用（包括配件及抽灌费用）是需要考虑的一项重要指标。假设设计一条 L 英尺长的管道，抽灌量为 Q g/m。目标是求出管道直径 D（单位 in），使其年度抽灌费用最低。考虑采用标准的碳钢泵，其年度抽灌费用可表示为[①]：

$$f(D) = 0.45L + 0.245LD^{1.5} + 325(hp)^{0.5} + 61.6(hp)^{0.925} + 102 \tag{13.22}$$

其中，

$$hp = 4.4 \times 10^{-8}\frac{LQ^3}{D^5} + 1.92 \times 10^{-9}\frac{LQ^{2.68}}{D^{4.68}} \tag{13.23}$$

下面求管长 1000 英尺、液体输送速率为 20 g/m 时，管道直径为多少才使其费用最低。首先创建函数 **PipeLineCost**，然后，用下面的程序可求出管道费用最低时的管道直径：

```
function PipeLine
L = 1000; Q = 20;
[D, fD] = fminbnd(@PipeLineCost, 0.25, 6, [ ], L, Q)

function f = PipeLineCost(D, L, Q)
hp = 4.4 * 10^( - 8) * L * Q^3/D^5 + (1.92 * 10^( - 9) * L * Q^2.68)/(D^4.68);
f = 0.45 * L + 0.245 * L * D^1.5 + 325 * hp^0.5 + 61.6 * hp^0.925 + 102;
```

执行结果为 $D = 1.1173$ in，$fD = f(D) = 1003$ \$/年。

例 13.11 封闭盒

如图 13.5 所示，在一张 90 cm × 90 cm 的硬纸板四角切下四个正方形，剩余部分做成一个封闭盒，求体积 V 最大时盒的尺寸。假定盒高 x 的下限及上限值分别为 8 cm 和 12 cm。设 $y = 90 - 3x$，封闭盒的体积为：

$$\begin{aligned} &\text{maximize}\, V(x) = xy^2 = x(90 - 3x)^2 \\ &\text{s.t.:} \quad 8 \leqslant x \leqslant 12 \end{aligned} \tag{13.24}$$

程序为：

```
L = 90;
Volume = inline(' - (L - 3 * x)^2 * x', 'x', 'L');
[x, V] = fminbnd(Volume, 8, 12, [ ], L)
```

执行程序得 $x = 10$ cm 及 $V = 36\,000$ cm^3。

① 参见 G. V. Reklaitis 等，出处同上，第 66 ~ 67 页。

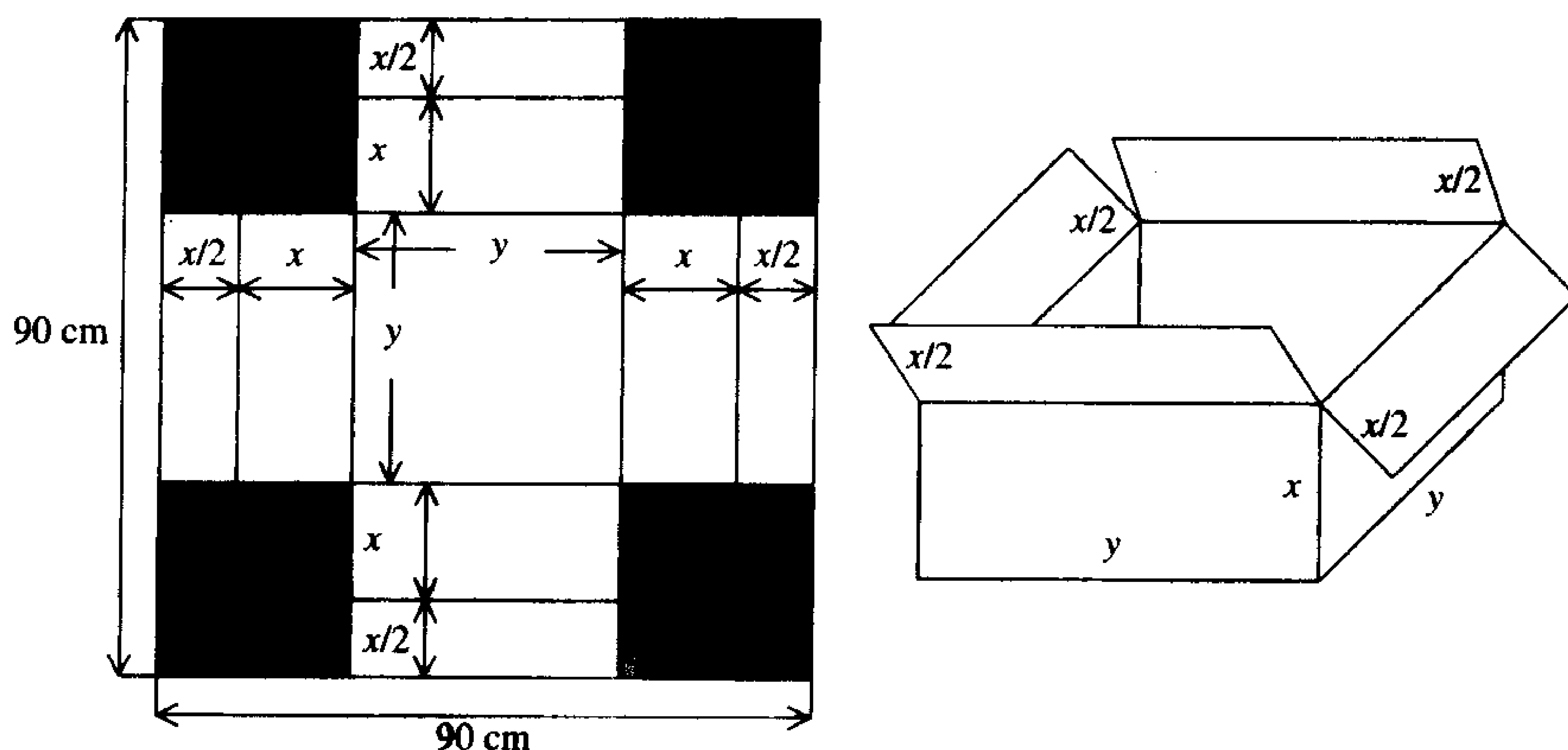


图 13.5　封闭盒的制作

13.4.2　多变量约束优化

多变量约束优化是解决非线性多变量约束问题的一种优化方法,由 fmincon 函数实现。约束条件有等式约束和不等式约束,目标函数和约束函数中有一个或多个为非线性函数。非线性多变量约束优化问题可做如下描述:

$$\begin{aligned}&\underset{x}{\text{minimize}}\quad f(x)\\&\text{s.t.}: Ax\leqslant b(\text{线性不等式约束})\\&\qquad A_{eq}x=b_{eq}(\text{线性等式约束})\\&\qquad C(x)\leqslant 0(\text{非线性不等式约束})\\&\qquad C_{eq}(x)=0(\text{非线性等式约束})\\&\qquad lb\leqslant x\leqslant ub,\cdots\end{aligned}\tag{13.25}$$

fmincon 函数的调用格式如下:

```
[xopt,fxopt] = fmincon(@UserFunction,x0,A,b,Aeq,beq,lb,ub,
                       @NonLinConsrtr,options,p1,p2,…)
```

其中,$xopt=x_{opt}$为 x 的最优解,$fxopt=f(x_{opt})$和 **UserFunction** 为求解目标函数的函数文件名,**UserFunction** 中必须给出输出量,其顺序与 fmincon 的帮助文件中定义的顺序一致。参数 $\boldsymbol{x}0$ 为向量的起始值,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束条件的系数,矩阵 $\boldsymbol{A}_{eq}$和向量 $\boldsymbol{b}_{eq}$为等式约束条件的系数。lb 和 ub 分别为 x 的下界和上界,*options* 为 optimset 定义的参数赋值,$p1$,$p2$ 等为传递给 **UserFunction** 和 **NonLinConstr** 的附加参数,**NonLinConstr** 为一函数名,该函数定义了非线性约束条件,其顺序与 fmincon 帮助文件中定义的顺序一致。传递给 **UserFunction** 和 **NonLinConstr** 的参数 $p1$,$p2$ 等必须一致,即使只用到一个函数时也必须如此。参数 lb,ub,*options*,A,b,A_{eq}和 b_{eq},如果没有定义,可用“[ ]”代替。

下面说明 fmincon 的用法。

例 13.12　两杆桁架

如图 13.6 所示的两杆桁架，目标函数为求两根木条 AC 和 BC 构成的架的最小体积。该函数有三个变量，x_1，x_2 和 y。x_1 和 x_2 分别为 AC 和 BC 交叉形成的横断面的两条边，y 为 C 点到水平面的垂直距离。另外还有几个约束条件，两杆的张应力限制为允许值 $\sigma = 10^5$ kPa，y 值介于 1 米和 3 米之间，x_1 和 x_2 为非负值。该优化问题可描述如下①：

$$\text{minimize } f_{\text{volume}} = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}$$

s.t.:

$$\begin{aligned} g_1 &: (20\sqrt{16+y^2})/(yx_1) - \sigma \leq 0 \\ g_2 &: (80\sqrt{1+y^2})/(yx_2) - \sigma \leq 0 \\ & 1 \leq y \leq 3 \\ & (x_1, x_2) \geq 0 \end{aligned} \tag{13.26}$$

因此，$A = b = A_{eq} = b_{eq} = C_{eq} = 0$，$lb = [1,0,0]$，$ub = [3,\infty,\infty]$。

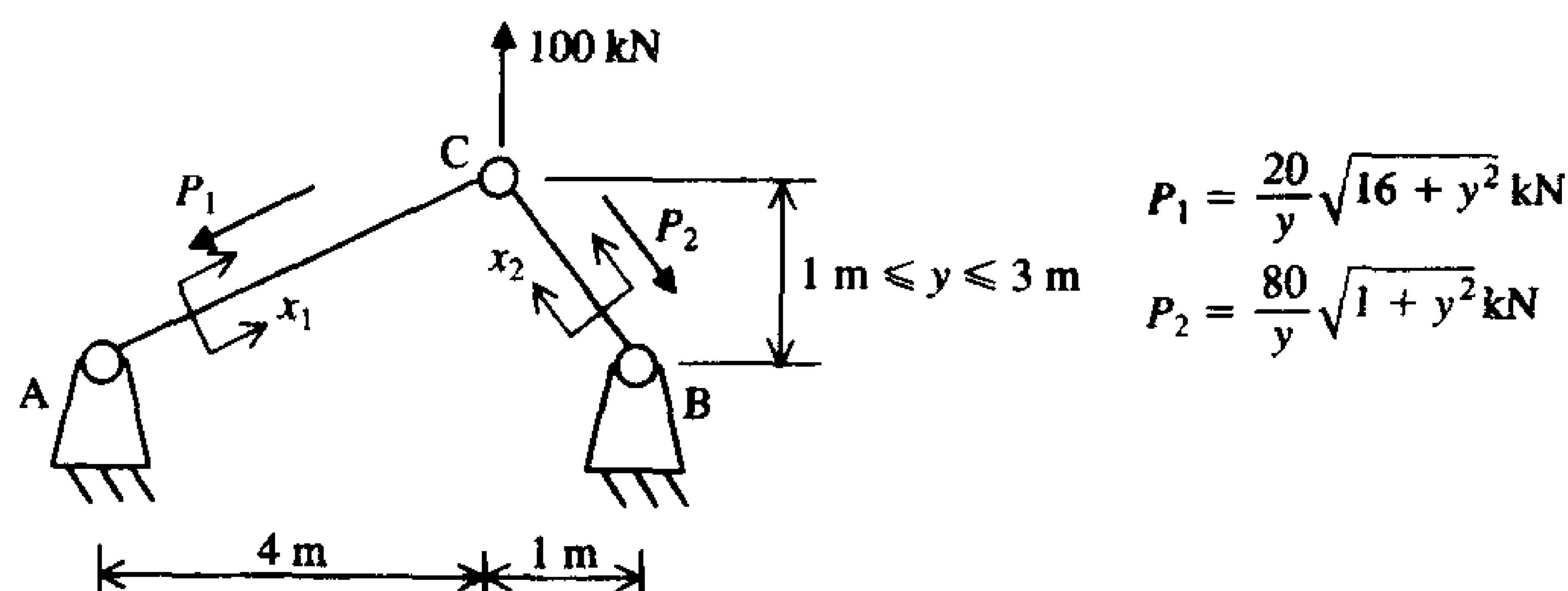


图 13.6　两杆桁架：x_1 和 x_2 为构成横截面的边

程序如下：

```
function TwoBarTruss
x0 = [1,1,1]; sigma = 10^5;
lb = [1 0 0]; ub = [3,inf,inf];
options = optimset('LargeScale','off');
[x,f] = fmincon(@TrussNonLinF,x0,[],[],[],[],lb,ub,...
                @TrussNonLinCon,options,sigma)
function f = TrussNonLinF(x,sigma)
y = x(1); x1 = x(2); x2 = x(3);
f = x1 * sqrt(16 + y^2) + x2 * sqrt(1 + y^2);

function[C,Ceq] = TrussNonLinCon(x,sigma)
y = x(1); x1 = x(2); x2 = x(3);
C(1) = 20 * sqrt(16 + y^2) - sigma * y * x1;
C(2) = 80 * sqrt(1 + y^2) - sigma * y * x2;
Ceq = [];
```

执行结果为 $x(1) = y = 1.9519$，$x(2) = x_1 = 0.0005$，$x(3) = x_2 = 0.0009$，$f = 0.004$。本例中关闭了默认的大规模算法，采用了中规模优化算法(参见例 13.4)。

① 参见 U. Kirsch, *Optimal Structural Design*, McGraw-Hill, New York, NY, 1981 年。

例 13.13 螺旋形受压弹簧

很多机械装置中都用到了螺旋形受压弹簧,用来施力、提供弹性以及存储或吸收能量。在设计这种螺旋形受压弹簧时,需要考虑诸如疲劳性、易弯曲性、摇摆、变形之类的一些要求。要满足这些不同的机械要求,可用最优化方法实现求解。描述该问题的公式如下文所述,其中包括一个设计目标函数、两个设计变量、七个约束条件及变量的上界和下界。

设计目标定为求安全系数倒数的极小值,等价于求安全系数的极大值。参考图 13.7,两个设计变量为 c 和 d,其中 $c=D/d$ 为弹簧指数,D 为弹簧的中径,d 为金属丝直径,设计目标为考虑疲劳或易弯曲指标时,求解安全系数倒数的极小值。考虑疲劳指标时,安全系数 SF_f 的倒数由下式确定:

$$\frac{1}{SF_f}=\frac{\tau_a}{S_{ns}}+\frac{\tau_m}{S_{us}} \tag{13.27}$$

式中,τ_a 和 τ_m 分别为剪应力的交变量和平均量,S_{ns} 为弹簧的材料疲劳强度,S_{us} 为其极限强度。考虑易弯曲指标时,安全系数 SF_y 的倒数由下式确定:

$$\frac{1}{SF_y}=\frac{\tau_a+\tau_m}{S_{ys}} \tag{13.28}$$

式中,S_{ys} 为剪切弯曲强度。如果满足下式,

$$\frac{\tau_a}{\tau_m}\geqslant\frac{S_{ns}(S_{ys}-S_{us})}{S_{us}(S_{ns}-S_{ys})} \tag{13.29}$$

则目标函数将为式(13.27)给出的考虑疲劳时的安全系数的倒数公式,否则目标函数为式(13.28)给出的考虑易弯曲时的安全系数的倒数公式。剪应力的平均量和交变量分别由下式确定:

$$\begin{aligned}\tau_a&=\frac{8F_acK_w}{\pi d^2}\\ \tau_m&=\frac{8F_mcK_w}{\pi d^2}\end{aligned} \tag{13.30}$$

式中,

$$\begin{aligned}K_w&=\frac{4c-1}{4c+4}+\frac{0.615}{c}\\ F_a&=(F_U-F_L)/2\\ F_m&=(F_U+F_L)/2\end{aligned} \tag{13.31}$$

其中,F_U 和 F_L 分别为施加于弹簧轴心的压力的上下限,F_a 和 τ_a 分别为交变力和剪应力,F_m 和 τ_m 分别为平均力和剪应力,K_w 为弹簧的弯曲剪切效应和直接剪切效应的 Wahl 应力系数。

下面给出了螺旋形受压弹簧的全部设计优化公式,考虑疲劳指标时设计目标为式(13.32)表示的形式:

$$\text{minimize}\quad\frac{1}{SF_f}=\frac{\tau_a}{S_{ns}}+\frac{\tau_m}{S_{us}} \tag{13.32}$$

考虑易弯曲指标时为式(13.26)所示的形式:

$$\text{minimize}\quad\frac{1}{SF_y}=\frac{\tau_a+\tau_m}{S_{ys}} \tag{13.33}$$

上述目标函数的选择取决于是否满足式(13.29)表示的条件。目标函数的约束条件为：

$$
\begin{aligned}
&g_1: K_1 d^2 - c \leqslant 0 && \text{摇摆约束}\\
&g_2: K_2 - c^5 \leqslant 0 && \text{变形约束}\\
&g_3: K_3 c^3 - d \leqslant 0 && \text{弹簧最小圈数约束}\\
&g_4: K_4 d^2 c^{-3} + K_8 d - 1 \leqslant 0 && \text{压缩长度约束}\\
&g_5: K_5 (cd + d) - 1 \leqslant 0 && \text{线圈的最大直径约束}\\
&g_6: c^{-1} + K_6 d^{-1} c^{-1} - 1 \leqslant 0 && \text{线圈的最小直径约束}\\
&g_7: K_7 c^3 - d^2 \leqslant 0 && \text{撞击余量约束}
\end{aligned}
\tag{13.34}
$$

其中，

$$
\begin{aligned}
&K_1 = \frac{G f_r \Delta}{112\,800(F_U - F_L)} && K_2 = \frac{G F_U (1 + A)}{22.3 k^2} && K_3 = \frac{8 k N_{\min}}{G}\\
&K_4 = \frac{G(1 + A)}{8 k L_m} && K_5 = \frac{1}{OD} && K_6 = ID\\
&K_7 = \frac{0.8(F_U - F_L)}{AG} && K_8 = \frac{Q}{L_m} && k = \frac{F_U - F_L}{\Delta}\\
&S_{ns} = C_1 d^{d_1} \overline{NC}^{B_1} && S_{us} = C_2 d^{A_1} && S_{ys} = C_3 d^{A_1}
\end{aligned}
\tag{13.35}
$$

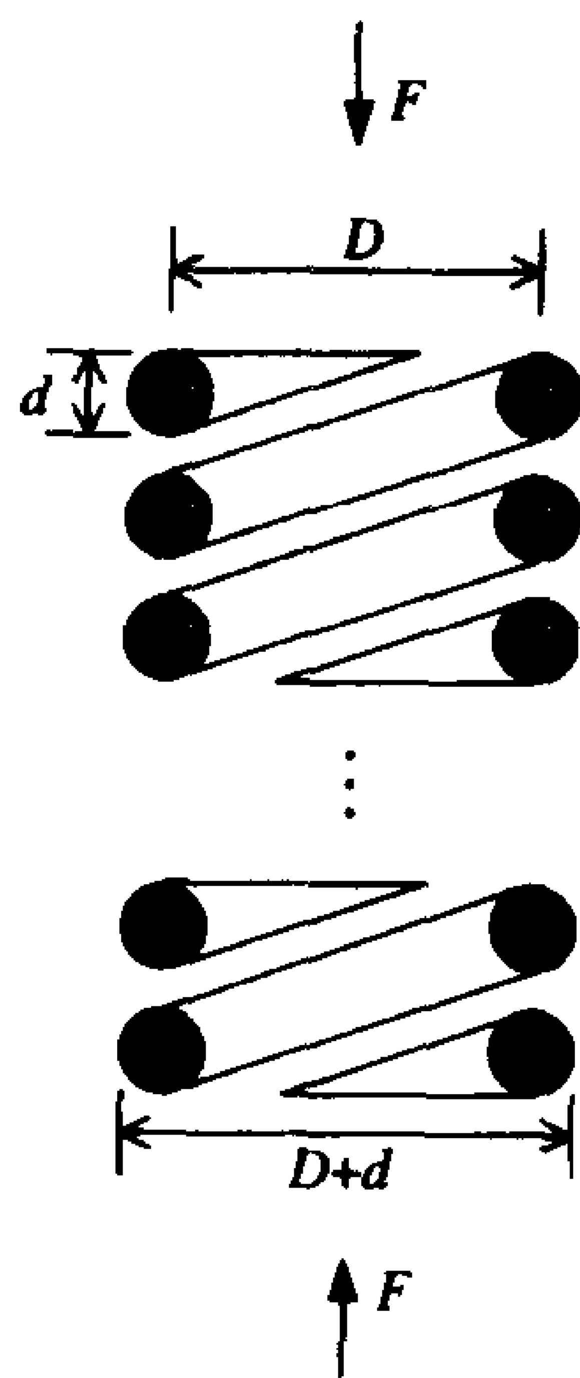


图 13.7　螺旋形受压弹簧

下面给出上述定义中用到的每个量的假定值：

$A = 0.4$	距离常数(无量纲)
$f_r = 500$ Hz	自然频率的最小允许值
$G = 11.5 \times 10^6$ psi	钢的剪切模量
$ID = 0.75$ in	弹簧内径的最小允许值
$OD = 1.5$ in	弹簧外径的最大允许值
$N_{\min} = 3$	最小线圈数

$L_m = 1.25$ in	弹簧承受最大负荷时的最大长度
$Q = 2$	端簧圈数
$\overline{NC} = 10^6$ cycles	失效周数
$\Delta = 0.25$ in	弹簧挠度

假定弹簧材料为钢丝,其 $A_1 = 0.14, B_1 = -0.2137, C_1 = 630\,500, C_2 = 160\,000, C_3 = 86\,550$。则弹簧指数 c 和线圈直径 d 的下界和上界分别为:

$$4 \leqslant c \leqslant 20$$

和

$$0.004 \leqslant d \leqslant 0.25$$

在此基础上创建三个函数,第一个为 **SpringParameters** 函数,用来计算不同的弹簧常数,后两个函数要求这些常数。第二个函数为 **SpringNLConstr**,用来计算非线性约束条件。第三个函数为 **SpringObjFunc**,用来计算目标函数:

```
function HelicalSpring
A = 0.4;FL = 15;FU = 30;G = 11.5 * 10^6;
fr = 500;ID = 0.75;OD = 1.5;Lm = 1.25;
NC = 10^6;Nmin = 3;Q = 2;Delta = 0.25;
A1 = -0.14;B1 = -0.2137;
C1 = 630500;C2 = 160000;C3 = 86550;
[K,Fa,Fm] = SpringParameters(A,FL,FU,G,fr,ID,OD,Lm,Nmin,Q,Delta);
x0 = [10,10];lb = [4,0.004];ub = [20,0.25];
options = optimset('LargeScale','off');
[x,f] = fmincon(@SpringObjFunc,x0,[],[],[],[],lb,ub,...
    @SpringNLConstr,options,K,Fa,Fm,NC,A1,B1,C1,C2,C3)
SafetyFactor = 1/f

function[K,Fa,Fm] = SpringParameters(A,FL,FU,G,fr,ID,OD,Lm,Nmin,Q,Delta)
Fa = (FU - FL)/2;
Fm = (FU + FL)/2;
k = (FU - FL)/Delta;
K(1) = G * fr * Delta/(112800 * (FU - FL));
K(2) = G * FU * (1 + A)/(22.3 * k^2);
K(3) = 8 * k * Nmin/G;
K(4) = G * (1 + A)/(8 * k * Lm);
K(5) = 1/OD;
K(6) = ID;
K(7) = 0.8 * (FU - FL)/(A * G);
K(8) = Q/Lm;

function[C,Ceq] = SpringNLConstr(x,K,Fa,Fm,NC,A1,B1,C1,C2,C3)
c = x(1);d = x(2);
C(1) = K(1) * d^2 - c;
C(2) = K(2) - c^5;
C(3) = K(3) * c^3 - d;
C(4) = K(4) * d^2/c^3 + K(8) * d - 1;
C(5) = K(5) * (c * d + d) - 1;
C(6) = 1/c + K(6)/c/d - 1;
C(7) = K(7) * c^3 - d^2;
Ceq = [];
```

```
function f = SpringObjFunc(x,K,Fa,Fm,NC,A1,B1,C1,C2,C3)
c = x(1);d = x(2);
Sns = C * d^A1 * NC^B1;
Sus = C2 * d * A1;
Sys = C3 * d^A1;
Kw = (4 * c - 1)/(4 * c + 4) + 0.615/c;
Temp = 8 * c * Kw/(pi * d^2);
TauA = Fa * Temp;
TauM = Fm * Temp;
Ratio = TauA/TauM;
SS = Sns * (Sys - Sus)/(Sus * (Sus - Sys));
if(Ratio - SS) > = 0
  f = TauA/Sns + TauM/Sus;
else
  f = (TauA + TauM)/Sys;
end
```

执行结果为 $x(1)=c=8.4987, x(2)=d=0.100$，安全系数 = 1.8408。本例中默认的大规模算法被关闭，使用了中规模算法。

例 13.14　齿轮减速箱

如图 13.8 所示，由两个齿轮组成齿轮组，其中一个为主齿轮，另一个为副齿轮。要求设计的齿轮组两轮子的体积及相关的轴体积均最小。该问题中有以下七个设计变量：

x_1 = 齿宽
x_2 = 模
x_3 = 副齿轮的齿数
x_4 = 轴承 1 两端的距离
x_5 = 轴承 2 两端的距离
x_6 = 轴 1 的直径
x_7 = 轴 2 的直径

其中，每个变量的上下限值为：

$$\begin{gathered}2.6 \leqslant x_1 \leqslant 3.6 \\ 0.7 \leqslant x_2 \leqslant 0.8 \\ 17 \leqslant x_3 \leqslant 28 \\ 7.3 \leqslant x_4 \leqslant 8.3 \\ 7.3 \leqslant x_5 \leqslant 8.3 \\ 2.9 \leqslant x_6 \leqslant 3.9 \\ 5.0 \leqslant x_7 \leqslant 5.5\end{gathered} \tag{13.36}$$

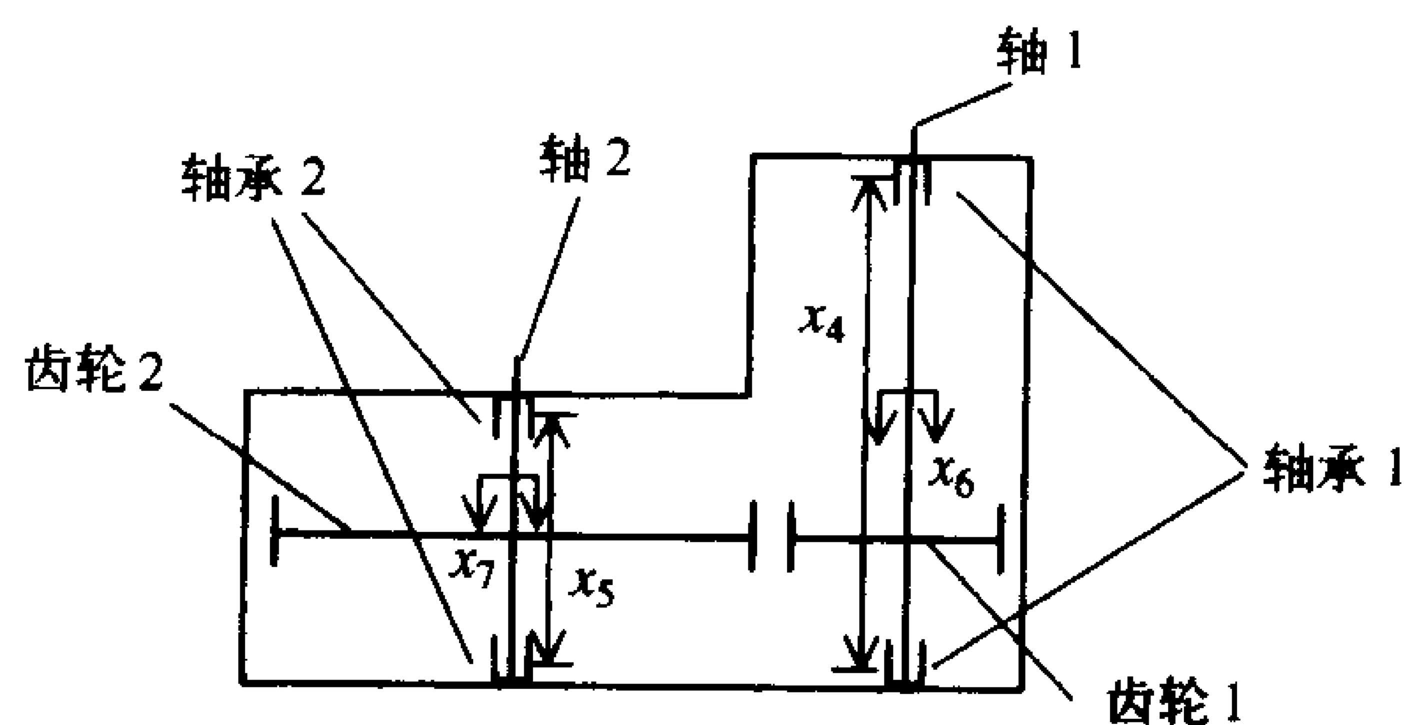


图 13.8　齿轮减速箱

设计目标为求所有轴总体积的最优解,该问题描述如下:

$$\text{minimize } f = 0.7854x_1x_2^2(3.3333x_3^2 + 14.933x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \tag{13.37}$$

上式中所有变量的单位为 cm。其约束条件为①:

$g_1: \dfrac{1}{(x_1x_2^2x_3)} - \dfrac{1}{27} \leqslant 0$ 齿轮的弯曲应力约束

$g_2: \dfrac{1}{(x_1x_2^2x_3^2)} - \dfrac{1}{397.5} \leqslant 0$ 齿轮的接触应力约束

$g_3: \dfrac{x_4^3}{(x_2x_3x_6^4)} - \dfrac{1}{1.93} \leqslant 0$ 轴 1 的偏差约束

$g_4: \dfrac{x_5^3}{(x_2x_3x_7^4)} - \dfrac{1}{1.93} \leqslant 0$ 轴 2 的偏差约束

$g_5: \dfrac{1}{0.1x_6^3}\sqrt{\left(\dfrac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6} - 1100 \leqslant 0$ 轴 1 应力约束

$g_6: \dfrac{1}{0.1x_7^3}\sqrt{\left(\dfrac{745x_5}{x_2x_3}\right)^2 + 157.5\times10^6} - 850 \leqslant 0$ 轴 2 应力约束

$g_7: x_2x_3 - 40 \leqslant 0$ 间隔限制约束

$g_8: 5x_2 - x_1 \leqslant 0$ 间隔限制约束

$g_9: x_1 - 12x_2 \leqslant 0$ 间隔限制约束

$g_{10}: 1.9 - x_4 + 1.5x_6 \leqslant 0$ 轴规格约束

$g_{11}: 1.9 - x_5 + 1.1x_7 \leqslant 0$ 轴规格约束

$g_1 \sim g_7$ 为非线性不等式约束条件,$g_8 \sim g_{11}$ 为线性不等式约束条件。由线性不等式约束条件可得:

$$A = \begin{bmatrix} -1 & 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & -12 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1.1 \end{bmatrix}$$

$$b = [0 \quad 0 \quad -1.9 \quad -1.9]'$$

且 $\boldsymbol{A}_{eq} = \boldsymbol{b}_{eq} = \boldsymbol{C}_{eq} = 0$。由式(13.36)得出 $\boldsymbol{L}_{bound}$ = [2.6,0.7,17,7.3,7.3,2.9,5], U_{bound} = [3.6,0.8,28,8.3,8.3,3.9,5.5]。

程序及两个子函数定义如下:

```
function Gears
x0 = [2.6,0.7,17,7.3,7.3,2.9,5];
lb = [2.6,0.7,17,7.3,7.3,2.9,5];
ub = [3.6,0.8,28,8.3,8.3,3.9,5.5];
```

① 参见 J. Golinski, "Optimum Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods", *Journal of Mechanisms*, 5(1970), 第 287 ~ 309 页。

```
A = aeros(4,7);
A(1,1) = - 1;A(1,2) = 5;
A(2,1) = 1;A(2,2) = - 12;
A(3,4) = - 1;A(3,6) = 1.5;
A(4,5) = - 1;A(4,7) = 1.1;
b = [0 0  - 1.9  - 1.9];
options = optimset('LargeScale','off');
[x,f] = fmincon(@GearObjFunc,x0,A,b,[],[],lb,ub,@GearNonLinConstr,..
                options)

function f = GearObjFunc(x)
f = 0.7854 * x(1) * x(2)^2 * (3.3333 * x(3)^2 + 14.9334 * x(3) - 43.0934)...
                - 1.508 * x(1) * (x(6)^2 + x(7)^2) + 7.477 * (x(6)^3 + x(7)^3)...
                + 0.7854 * (x(4) * x(6)^2 + x(5) * x(7)^2);

function[C,Ceq] = GearNonLinConstr(x)
C(1) = 1/(x(1) * x(2)^2 * x(3)) - 1/27;
C(2) = 1/(x(1) * x(2)^2 * x(3)^2) - 1/397.5;
C(3) = x(4)^3/(x(2) * x(3) * x(6)^4) - 1/1.93;
C(4) = x(5)^3/(x(2) * x(3) * x(7)^4) - 1/1.93;
C(5) = sqrt((745 * x(4)/(x(2) * x(3)))^2 + 16.9 * 10^6)/(0.1 * x(6)^3) - 1100;
C(6) = sqrt((745 * x(5)/(x(2) * x(3)))^2 + 157.5 * 10^6)/(0.1 * x(7)^3) - 850;
C(7) = x(2) * x(3) - 40;
Ceq = [];
```

执行结果为：$[x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [3.500, 0.700, 17.000, 7.300, 7.7153, 3.3502, 5.2867]$及$f = 2994.3$。

13.4.3　二次规划

二次规划指一类特殊的约束优化问题，其目标函数为二次多项式，约束条件为线性约束，即：

$$\begin{aligned} \underset{x}{\text{minimize}} f &= 0.5x^T Hx + c^T x \\ \text{s.t.}: Ax &\leqslant b \\ A_{eq}x &= b_{eq} \\ lb \leqslant x &\leqslant ub \end{aligned} \tag{13.38}$$

其中 $\boldsymbol{H}$,$\boldsymbol{A}$ 和 $\boldsymbol{A}_{eq}$ 为矩阵，$\boldsymbol{b}$,$\boldsymbol{b}_{eq}$,$\boldsymbol{c}$,$\boldsymbol{x}$,$\boldsymbol{lb}$ 和 $\boldsymbol{ub}$ 为列向量，解决该类问题的 MATLAB 函数为：

```
[xopt,fopt] = quadprog(H,c,A,b,Aeq,beq,lb,ub,x0,options,p1,p2,...)
```

其中 $xopt = x_{opt}$ 为 x 的优化值，$fopt = f(x_{opt})$，对称矩阵 $\boldsymbol{H}$ 和向量 $\boldsymbol{c}$ 为二次目标函数 f 的系数，矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为非线性不等式约束条件的系数，矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 为线性等式约束条件的系数，向量 $\boldsymbol{lb}$ 和 $\boldsymbol{ub}$ 定义了设计变量 x 的下界和上界。向量 $\boldsymbol{x0}$ 为起点。*options* 定义 Optimset 中参数的值。

下面描述了 quadprog 的用法。

例 13.15　生产计划问题

例 13.2 中讨论的生产计划问题在这里稍微有所改变。现在假定产品 A 和 B 的利润与其数量之间为一函数关系。对产品 A 来说，单位产品的利润(美元)按下式计算：

$$4 + 2x_1 + 3x_2$$

产品 B 的单位产品利润为:

$$5 + 5x_1 + 4x_2$$

其余条件与例 13.2 中的相同,由此得出最优目标函数为:

$$\begin{aligned} f &= -(4 + 2x_1 + 3x_2)x_1 - (5 + 5x_1 + 4x_2)x_2 \\ &= -(4 + 2x_1)x_1 - (5 + 4x_2)x_2 - 8x_1x_2 \end{aligned}$$

或

$$f = \frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} -4 & -8 \\ -8 & -8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -4 & -5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{13.39}$$

程序如下:

```
H=[-4,-8;-8,-8];
c=[-4;-5];
A=[1,1;1.25,0.75;0,1];
b=[200;200;150];
lb=zeros(2,1);
options=optimset('LargeScale','off');
[xopt,fopt]=quadprog(H,c,A,b,[],[],lb,[],[],options)
```

执行结果为:$xopt(1) = x_1 = 50$,$xopt(2) = x_2 = 150$,$-fopt = -f(x_{opt}) = 155\,950$。

13.4.4 半无穷约束优化

利用半无穷约束优化对以下优化问题求解:

$$\begin{aligned} &\underset{x}{\text{minmize}} f(x) \\ &\text{s.t.}: Ax \leqslant b \\ &A_{eq}x = b_{eq} \\ &C(x) \leqslant 0 \\ &C_{eq}(x) = 0 \\ &K_1(x, w_1) \leqslant 0 \\ &K_2(x, w_2) \leqslant 0 \\ &\vdots \\ &K_n(x, x_n) \leqslant 0 \\ &\forall (w_1, \cdots, w_n) \end{aligned} \tag{13.40}$$

其中,$\boldsymbol{x}$ 为设计向量变量,f 为标量目标函数,$\boldsymbol{A}$ 为表示线性不等式约束的向量,$\boldsymbol{b}$ 为线性不等式约束条件向量,$\boldsymbol{A}_{eq}$ 为表示线性等式约束的向量,$\boldsymbol{b}_{eq}$ 为线性等式约束向量,$\boldsymbol{C}$ 为表示非线性不等式约束条件的向量算子,$\boldsymbol{C}_{eq}$ 为非线性等式约束向量算子。$\boldsymbol{K}_n(x, w_n)$ 为半无穷函数的向量(或矩阵),是 $\boldsymbol{x}$ 向量和 $\boldsymbol{w}_n$ 的函数。其中,自由变量表示一个解的范围,在此范围内求解设计变量 x 的值。$\forall(w_1, \cdots, w_n)$ 意为自由变量取其相应范围内的所有值,其中 $\forall$ 是"全部"的意思。例如在传热问题中,w_n 可以表示温度的变化范围,在振动问题中可表示频率范围。在变化范围内所有的温度值或频率值必须满足 K_n 约束条件。向量 $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n$ 的长度最多为 2。

MATLAB 函数为：

```
[xopt,fxopt] = fseminf(@UserFunction,x0,n,@SemiConstr,…
                       A,b,Aeq,Beq,lb,ub,options,p1,p2,…)
```

其中，*xopt* = x_{opt}为 x 的最优解，*fopt* = $f(x_{opt})$，**UserFunction** 为计算标量函数 f 的函数文件名，$x0$ 参数为起始点，n 为式(13.40)所示的半无穷约束条件数。矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束系数，矩阵 $\boldsymbol{A}_{eq}$和向量 $\boldsymbol{b}_{eq}$为线性等式约束系数。*lb* 和 *ub* 分别定义了设计变量 x 的上界和下界，*options* 设置 optimset 中定义的参数，$p1$，$p2$ 等为传递给 **UserFunction** 和 **SemiConstr** 的附加参数。**UserFunction** 和 **SemiConstr** 的 $p1$，$p2$ 等即使有一个函数用到这些值，也必须一致。如果没有定义 *lb*，*ub* 和 *options*，用[ ]代替。对 $\boldsymbol{A}$，$\boldsymbol{b}$，$\boldsymbol{A}_{eq}$和 $\boldsymbol{b}_{eq}$也是如此。参数 **SemiConstr** 为一函数名，该函数定义了非线性约束条件，如下所示：

```
function[C,Ceq,K1,K2,…,Kn,s] = SemicConstr(x,s,p1,p2,…)
% Initial sampling interval
If isnan(s(1,1)),
  s = …% s has n rows and 2 columns
end
W1 = …% computes sample set
…
wn = …% computes sample set
K1 = …% 1st semi-infinite constraint at x and w
…
Kn  = …% nth semi-infinite constraint at x and w
C   = …% computes nonlinear inequalities
Ceq = …% computes nonlinear equalities
```

其中，$K1$，$K2$，…，Kn 分别为估计自由变量在抽样值范围内的半无穷约束条件，两列矩阵 $\boldsymbol{s}$ 各行元素表示相关值 w_1，…，w_n 的采样间隔，用于计算 $K1$，…，Kn，即 $\boldsymbol{s}$ 的第 i 行包含了计算 $\boldsymbol{K}_i$ 的采样间隔。$\boldsymbol{K}_i$ 是向量时，只用到 $s(i,1)$(第 2 列元素可均为 0)，$\boldsymbol{K}_i$ 是矩阵时，$s(1,2)$用于 $\boldsymbol{K}_i$ 行的采样，$s(i,1)$用于 $\boldsymbol{K}_i$ 列的采样。在第一次迭代时，$\boldsymbol{s}$ 置为 NaN，定义采样间隔的初始值。如果 C 和 C_{eq}有一项不存在，或两者都不存在，则将其置为[ ]。

下面说明函数 fseminf 的用法。

例 13.16　平面两连杆操纵问题

平面两连杆操纵器如图 13.9 所示。该操纵器能够定位到平面上一点，设计目标为求操纵器所覆盖的最大工作范围。两个设计向量 $\boldsymbol{a}$ 与 $\boldsymbol{b}$ 表示两连杆的长度，约束条件有关于比值 $\boldsymbol{a}/\boldsymbol{b}$ 的下界约束(G_1)与上界约束(G_2)，灵巧性量度系数的上界约束(K_1)，还有设计变量的下界与上界约束 a 和 b。灵巧性(dexterity)[①] 一词是指操纵器能够很容易地在其工作范围内沿任意方向移动、用力或扭转。雅可比(Jacobian)矩阵的条件系数 κ 用于度量操纵器的灵巧性，其值越接近整数 1 越好。

① 参见 C. Gosselin 和 J. Angeles,"A Global Performance Index for the Kinetic Optimization of Robotic Manipulators", *ASME Journal of Mechanical Design*, 113, (Sept. 1991), 第 222 页。

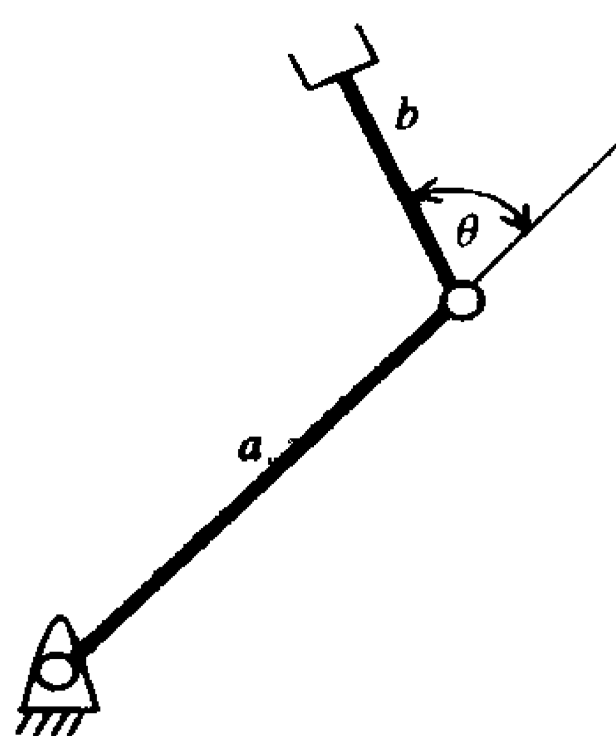


图 13.9　平面两连杆操纵器

下面是对该问题的描述：

$$\text{minimize} f(a,b) = -\pi[(a+b)^2 - (a-b)^2]$$

s.t.:

$$G_1: a/b \geqslant 1.1$$
$$G_2: a/b \leqslant 2$$
$$K_1: \kappa \leqslant 1.26$$
$$0.1 \leqslant a \leqslant 2$$
$$0.1 \leqslant b \leqslant 2$$
$$\forall \theta \in [100°, 150°] \tag{13.41}$$

其中，条件系数 κ 由下式确定：

$$\kappa = (a^2 + 2b^2 + 2ab\cos\theta)/2ab\cos\theta \tag{13.42}$$

半无穷约束条件为满足 $\theta \in [100°, 150°]$ 的变化范围，其采样间隔为 5°。程序及其子函数如下：

```
function PlanarManipulator
x0 = [1,1];
LBnd = [0.1,0.1]; UBnd = [2,2];
[x,fopt] = fseminf(@TwoLinkObjFunc,x0,1,@TwoLinkConstr,[],[],[],[],…
                    LBnd,UBnd,[])
text(120,1.17,'Initial')
text(113,1.05,'Optimum')
ylabel('Condition number \ kappa')
xlabel('\ theta(degrees)')
function[C,Ceq,K1,s] = TwoLinkConstr(x,s)
a = x(1); b = x(2);
if isnan(s(1,1))
  s = [5,0];
end
theta = (100:s(1,1):150) * pi/180;
K1 = (a^2 + 2 * b^2 + 2 * (a * b) * cos(theta))./(2 * (a * b) * sin(theta)) - 1.26;
C(1) = - a/b + 1.1;
C(2) = a/b - 2;
Ceq = [];
plot(theta * 180/pi,K1 + 1.26,'k')
hold on

function f = TwoLinkObjFunc(x)
a = x(1); b = x(2);
f = - pi * ((a + b)^2 - (a - b)^2);
```

程序执行结果形成图 13.10,并给出 $x(1)=a=2.0$ 及 $x(2)=b=1.4433$,其相应工作范围内的 $-f=36.2732$。图 13.10 中画出了每次迭代过程中条件系数 κ 与 θ 的关系曲线,以显示条件系数从其初始范围到最优范围的变化过程。从图中看出最优线 $\theta=136°$时,条件系数几乎等于 1,该角度即为考虑灵巧性时操作器的最佳形状。

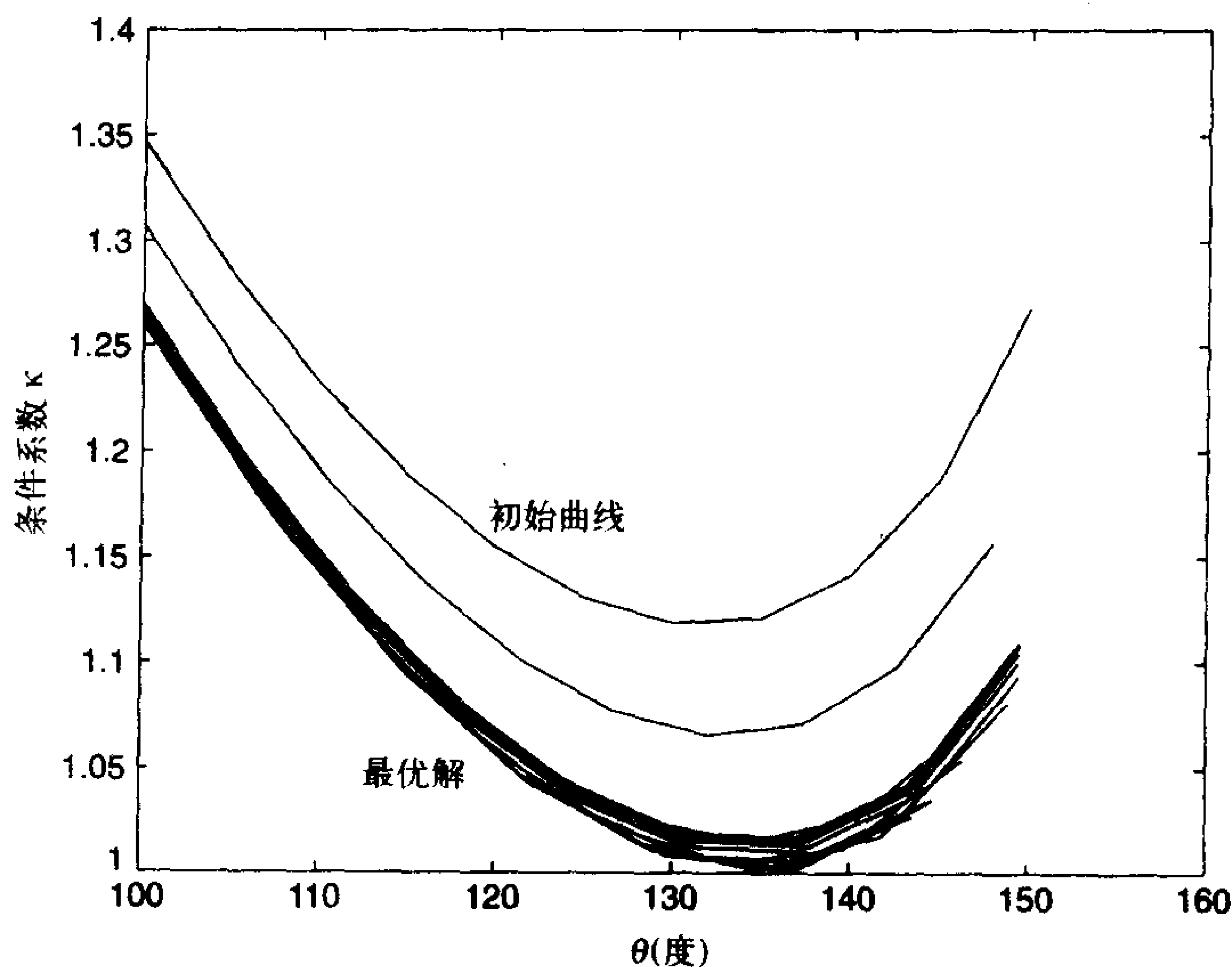


图 13.10 平面两连杆操纵器的条件系数从 a 和 b 的初值变化到最优解的渐近过程,条件系数 K_1 为 θ 的函数

13.5 多目标优化

多目标优化指设计目标多于一个的优化问题。该类问题中目标相互之间常常是矛盾的,其原因在于问题的内在属性。例如一拉紧构件问题中,两个设计目标分别为求重量的极小值和求应力的极小值。这两个目标之间相互矛盾,即若构件重量减轻,则应力会增加;反之亦然。在此类优化过程中,求得的点不一定同时满足所有目标,因此多目标问题中的"优化"通常指求得一个解,没有比之更好的解,从而使任何一个目标得到改善。这样一个解称做非劣点或有效点。多目标优化问题中可能存在许多非劣点,这些点一起形成一条"边界线"。图13.11所示为两变量两目标优化问题,图中画出了变量空间的可行域和目标空间的可行域,同时给出了 f_1 和 f_2 都取极小值时的边界线。如图中所示,目标空间中的可行域是由变量空间中的可行域曲线映射而来的。边界解集对应于能实现的"最优"解。解集存在一种折中方案,即该解可能使一个目标得到改善而另一目标将变坏。

多目标优化问题的最终解将根据决策者的选择从解集中确定。MATLAB 用于解决多目标优化问题的函数有两个:fminimax 和 fgoalattain。

fminimax 方法求解下列问题:

$$\min_{x}\max_{f}\ \{f_1,f_2,\cdots,f_m\}$$

s.t.:

$$
\begin{aligned}
Ax &\le b & &(\text{线性不等式约束})\\
A_{eq}x &= b_{eq} & &(\text{线性等式约束})\\
C(x) &\le 0 & &(\text{非线性不等式约束})\\
C_{eq}(x) &= 0 & &(\text{非线性等式约束})\\
lb \le x &\le ub
\end{aligned}
\tag{13.43}
$$

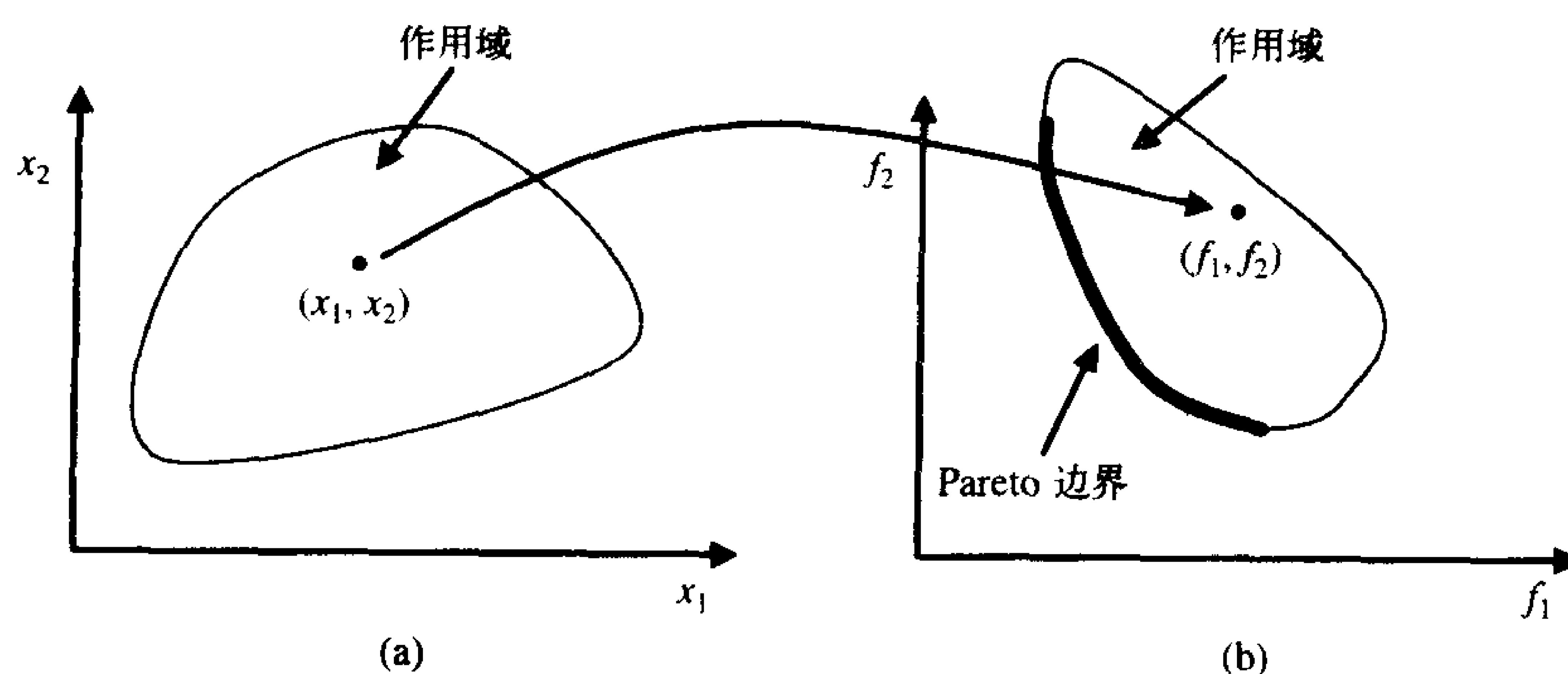


图 13.11 Pareto 边界的作用域:(a)变量空间;(b)目标空间

其中,x 为设计向量变量,$f_1, f_2, \cdots, f_m$ 为目标函数,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 存放线性不等式约束系数,矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 包含线性等式约束系数,C 包含非线性不等式约束,C_{eq} 包含非线性等式约束,lb 与 ub 分别为设计量的下界和上界。fminimax 方法使受约束目标函数的最坏值 x 逐次变小。

MATLAB 函数为:

```
[xopt,fxopt] = fminimax(@UserFunction,x0,A,b,Aeq,beq,lb,ub,…
                        @NonLinConstr,options,p1,p2,…)
```

式中,$xopt = x_{opt}$ 为 x 的最优值,$fxopt = f_i(x_{opt})$ 为计算目标函数的函数文件名,该函数必须以一定顺序生成输出量。**x0** 为起始值的向量,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束系数,矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 为等式约束系数,***lb*** 和 ***ub*** 分别为 x 的下界与上界向量,*options* 设置由 optimset 定义的参数值,$p1$,$p2$ 等为传递给 **UserFunction** 和 **NonLinConstr** 的附加参数。**NonLinConstr** 为一函数,该函数按照规定顺序定义了一组非线性约束条件,如下所示:

```
function[C,Ceq] = NonLinConstr(x,p1,p2,...)
```

传递给 **UserFunction** 和 **NonLinConstr** 的参数 $p1, p2, \cdots$ 必须保持一致,即使只有一个函数用到这些值或只用到其中一部分也必须如此。如果 *lb*, *ub* 和 *options* 等参数未定义,则用[]代替,$\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{A}_{eq}$ 和 $\boldsymbol{b}_{eq}$ 也是如此。

函数 fgoalattain 用于解以下多目标优化问题:

$$\underset{x,\gamma}{\text{maximize}}\ \gamma$$

s.t.:

$$
\begin{aligned}
& f_i(x) - \omega_i \gamma \leqslant (\text{goal})_i \qquad i = 1, \cdots, m \\
& Ax \leqslant b \qquad \text{（线性不等式约束）} \\
& A_{eq} x = b_{eq} \qquad \text{（线性等式约束）} \\
& C(x) \leqslant 0 \qquad \text{（非线性不等式约束）} \\
& C_{eq}(x) = 0 \qquad \text{（非线性等式约束）} \\
& lb \leqslant x \leqslant ub
\end{aligned} \tag{13.44}
$$

其中，γ 为可正可负的标量变量，f_i 为第 i 个目标函数，ω_i 和 $(\text{goal})_i$ 分别为第 i 个目标函数的加权系数和目标。加权系数控制未达到目标或超过目标的相对程度，$\omega_i \gamma$ 在公式中为一弱化量。例如，设所有加权系数均等于其初始目标，意味着期望未达到目标的百分比与超越目标的百分比相同。

MATLAB 函数为：

```
x = fgoalattain(@UserFunction, x0, Goal, Weight, A, B, Aeq, beq, lb, ub, …
                @NonLinConstr, options, p1, p2, …)
```

其中，向量 *Weight* 包含元素 ω_i，向量 *Goal* 包含元素 $(\text{goal})_i$，其余参数与 fminimax 中定义的相同。

下面的两个例子给出了 fminimax 和 fgoalattain 的应用。

例 13.17　振动台

在图 13.12 所示的系统中，安装有发动机的一梁型平台由三层材料构成。假定梁两端有简单的支撑物，由发动机引起的振动能够传递给梁。

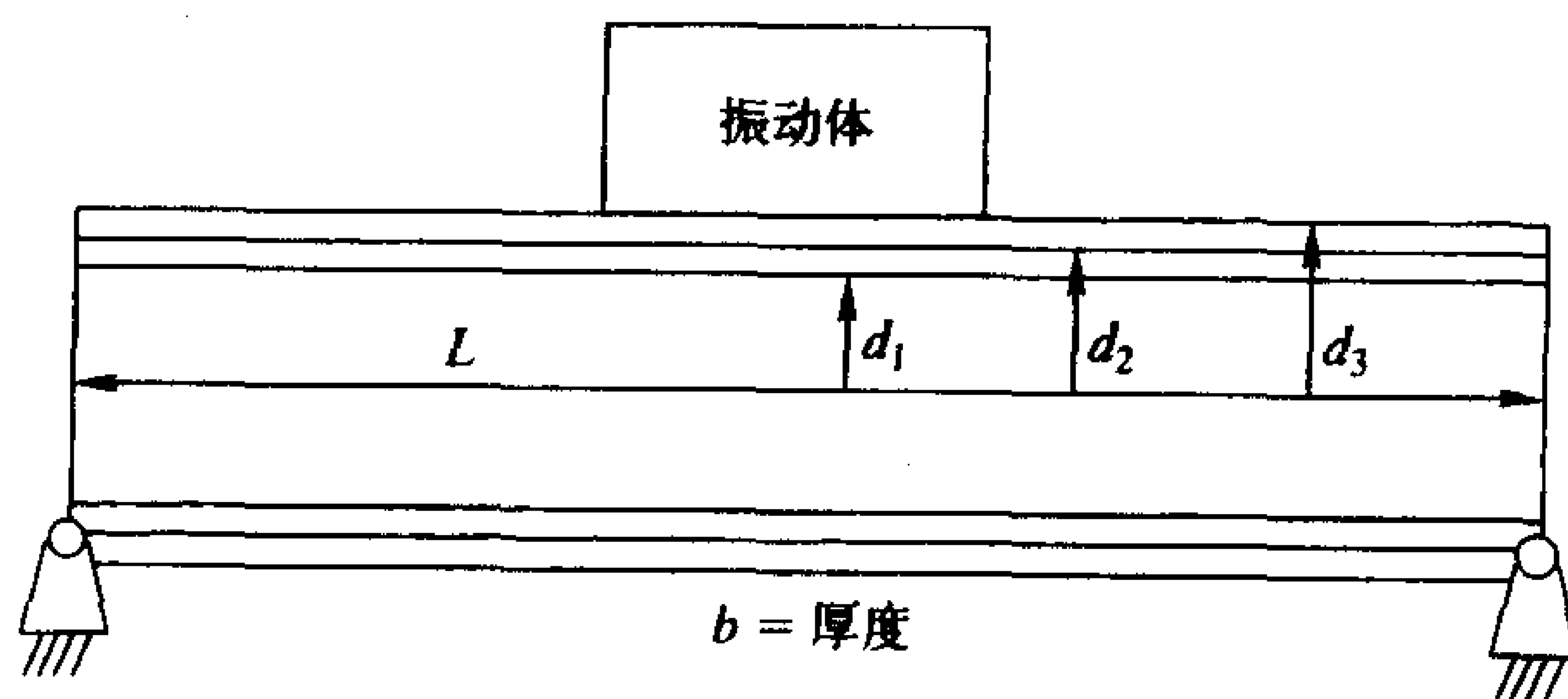


图 13.12　由多层材料构成的简单支撑振动台

该系统的设计目标为：

1. 对梁的基本自然频率的负值求极小值，用 f_1 表示。
2. 对梁所用材料的费用求极小值，用 f_2 表示。

约束条件包括：梁 g_1 的质量上限，第二层梁 g_2 和第三层梁 g_3 厚度的上限，还有设计变量的上下限。五个设计变量为梁的长度 L，宽度 b，到第一层 d_1 和第二层 d_2 接触面到梁的中心轴距离，以及到梁顶的距离 d_3。表 13.5 列出了三层材料中每层的质量密度 ρ，弹性

模量 E 及单位体积 c 的费用。问题描述如下[①]:

$$\text{minimize}\quad f_1(d_1,d_2,d_3,b,L) = -(\pi/2L^2)\sqrt{EI/\mu}$$

$$\text{minimize}\quad f_2(d_1,d_2,d_3,b) = 2b(c_1d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2))$$

其中,

$$EI = (2b/3)(E_1d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3))$$

$$\mu = 2b(\rho_1d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2))$$

s.t.:

$g_1: \mu L - 2800 \leqslant 0$　　梁的质量

$g_2: d_2 - d_1 - 0.15 \leqslant 0$　　层厚

$g_3: d_3 - d_2 - 0.01 \leqslant 0$　　层厚

$$0.05 \leqslant d_1 \leqslant 0.5$$

$$0.2 \leqslant d_2 \leqslant 0.5$$

$$0.2 \leqslant d_3 \leqslant 0.6$$

$$0.35 \leqslant b \leqslant 0.5$$

$$3 \leqslant L \leqslant 6$$

表 13.5　振动台的材料性能指标及费用

层 i	ρ_i(kg/m³)	E_i(N/m²)	c_i($/体积)
1	100	1.6×10^9	500
2	2770	70×10^9	1500
3	7780	200×10^9	800

由此看出 g_1 为非线性不等式约束,g_2 和 g_3 为线性不等式约束,因此:

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

$$b = [0.15 \quad 0.01]'$$

因为没有线性和非线性等式约束,所以 $C_{eq} = A_{eq} = b_{eq} = 0$。

为保持与计算函数有相同的数量级,设计目标换算为下式:

$$scaled_value = \frac{raw - good}{bad - good} \tag{13.45}$$

式中,*raw* 指函数实际值(换算前),*good* 指函数的理想值,*bad* 指非理想值。从式(13.45)可以看出,当 *raw* 与 *good* 相等时,$scaled_value = 0$;当 *raw* 与 *bad* 相等时,$scaled_value = 1$。下面的程序使用三个子函数:**BeamProperties**,其中,$[x_1, x_2, x_3, x_4, x_5] = [d_1, d_2, d_3, b, L]$。第二个函数 **VibPlatNLConstr** 计算非线性不等式约束;第三个函数 **VibPlatformobj** 计算目标函数。

① 修订自 A. Messac,"Physical Programming: Effective Optimization for Computational Design", *AIAA Journal*, 34(1), 1996 年,第 149 ~ 158 页。

程序为：

```
function VibratingPlatform
x0 = [0.3,0.35,0.4,5,0.4];
lb = [0.05,0.2,0.2,0.35,3];
ub = [0.5,0.5,0.6,0.5,6];
E = [1.6,70,200] * 10^9;
Rho = [100,2770,7780];
c = [500,1500,800];
good = [500,100];
A = [ - 1,1,0,0,0;0, - 1,1,0,0];
b = [0.15,0.01]';
for k = 1:5
  bad = [100 + k * 10 500 - k * 50];
  [xopt,fxopt] = fminimax(@VibPlatformObj,x0,A,b,[],[],lb,ub,...
                @VibPlatNLConstr,[],E,Rho,c,good,bad);
  for m = 1:2
    ff(m) = fxopt(m) * (bad(m) - good(m)) + good(m);
  end
  f1(k) = ff(1);
  f2(k) = ff(2);
end
[f2sort,indxf2] = sort(f2);
f1sort = f1(indxf2);
plot( - f1sort,f2sort,'ko - ');
xlabel('Negative frequency(Hz)');
ylabel('Cost($)');

function[EI,mu] = BeamProperties(x,E,Rho,c)
EI = (2 * x(4)/3) * (E(1) * x(1)^3 + E(2) * (x(2)^3 - x(1)^3) + E(3) * (x(3)^3 - x(2)^3));
mu = 2 * x(4) * (Rho(1) * x(1) + Rho(2) * (x(2) - x(1)) + Rho(3) * (x(3) - x(2)));

function[C,Ceq] = VibPlatNLConstr(x,E,Rho,c,good,bad)
[EI,mu] = BeamProperties(x,E,Rho,c);
C(1) = mu * x(5) - 2800;
Ceq = [];

function f = VibPlatformObj(x,E,Rho,c,good,bad)
[EI,mu] = BeamProperties(x,E,Rho,c);
f1 = pi/(2 * x(5)^2) * sqrt(EI/mu);
f(1) = (f1 - good(1))/(bad(1) - good(1));
f2 = 2 * x(4) * (c(1) * x(1) + c(2) * (x(2) - x(1)) + c(3) * (x(3) - x(2)));
f(2) = (f2 - good(2))/(bad(2) - good(2));
```

以上程序的执行结果为图 13.13 所示的边界线。

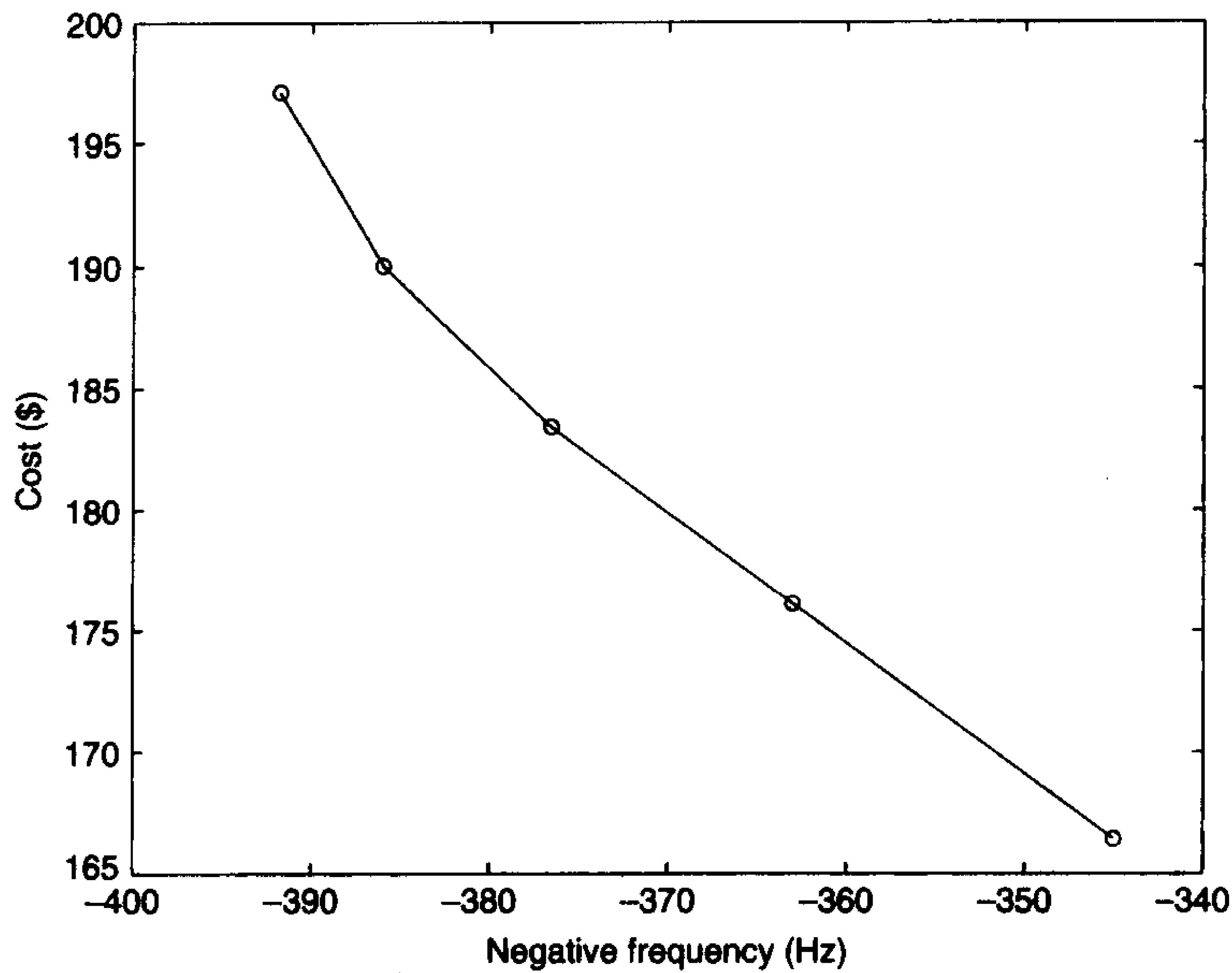


图 13.13 振动台的 Pareto 边界线

例 13.18 生产计划问题

再次考虑例 13.2 中的生产计划问题。现在介绍第二种设计目标,即对产品 A 的生产件数求极大值(或对其负值求极小值)。下面是其目标函数和约束条件:

$$\text{minimize } f_1(x_1, x_2) = -4x_1 - 5x_2$$

$$f_2(x_1) = -x_1$$

s.t.:

$$g_1: x_1 + x_2 \leqslant 200$$

$$g_2: 1.25x_1 + 0.75x_2 \leqslant 200$$

$$g_3: x_2 \leqslant 150$$

$$(x_1, x_2) \geqslant 0$$

因此,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1.25 & 0.75 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{b} = [200 \quad 200 \quad 150]'$$

可以得到其下限 $\boldsymbol{lb} = [0\ 0]$和上限 $\boldsymbol{ub} = [\infty, \infty]$。这里没有等式约束和非线性约束条件,所以 $\boldsymbol{C}_{eq} = \boldsymbol{A}_{eq} = \boldsymbol{b}_{eq} = \boldsymbol{C} = 0$。

程序如下:

```
A = [1 1;1.25 0.75;0 1];
b = [200,200,150]';
goal = [ - 950, - 50];x0 = [50,50];
lb = [0,0];ub = [inf,inf];
Weight = abs(goal);
options = optimset('GoalsExactAchieve',2);
```

```
ProdPlanObj = inline('[ -4 * x(1) - 5 * x(2), - x(1)]','x');
[x,fxopt] = fgoalattain(ProdPlanObj,x0,goal,Weight,A,b,[],[],…
                        lb,ub,[],options)
```

其中，参数 *GoalsExactAchieve* 值设为 2（独立变量数），这是一种正好达到目标的算法，既不过也不欠。如果用默认值，则解为 $x0$ 的一个函数，程序执行结果得到一个解 $[x_1, x_2] = [50,150]$ 及 *fxopt* $= [-950, -50]$。

13.6 应用函数一览

表 13.6 列出了本章介绍的优化工具箱中的函数及其说明。

表 13.6　第 13 章介绍的优化工具箱中的 MATLAB 函数

MATLAB 函数	说明
fgoalattain	多目标求解
fminbnd	求单变量函数在固定间隔内的最小值
fmincon	求多变量受约束非线性函数的最小值
fminimax	求最大值解
fminsearch	求多变量无约束函数的最小值
fminunc	求多变量无约束函数的最小值
fseminf	求多变量半无限约束非线性函数的最小值
linprog	求线性规划解
lsqcurvefit	用最小二乘法求非线性曲线拟合解
lsqnonlin	非线性最小二乘法的解
quadprog	求二次规划解

练习

13.1　如图 13.14 所示的机械装置，其中棒 OA 和 AC 的长度均为 $L_1 = 0.5$ m，CB 和 BD 的长度均为 $L_2 = 0.3$ m，O 点固定，C 点和 D 点分别钉在两滑块中，并且可沿水平方向无摩擦移动。该装置受三个外力作用，其中在 A 点和 B 点的垂直力分别为 $P_1 = 3$ kN和 $P_2 = 1$ kN。D 点有一水平方向力 $P = 3$ kN。假定未施力之前该装置中的四根棒沿水平线调成一直线。忽略该装置的重量，则该装置在施加外力时的平衡状态可通过求其势能（PE）函数的极小值得到：

$$\text{minimize PE} = -P_1 L_1 \sin\alpha - P_2 L_2 \sin\beta + P[L_1(1-\cos\alpha) + L_2(1-\cos\beta)]$$

（a）给出 PE 的等高线图 α 和曲面图 β，如图 13.15 所示。

（b）用函数 fminunc 计算并证实上述结果。初始点 $(\alpha, \beta) = (1,1)$ 弧度。

［答案：$\alpha = 0.4636$ rad，$\beta = 0.1651$ rad，PE $= -0.3789$。］

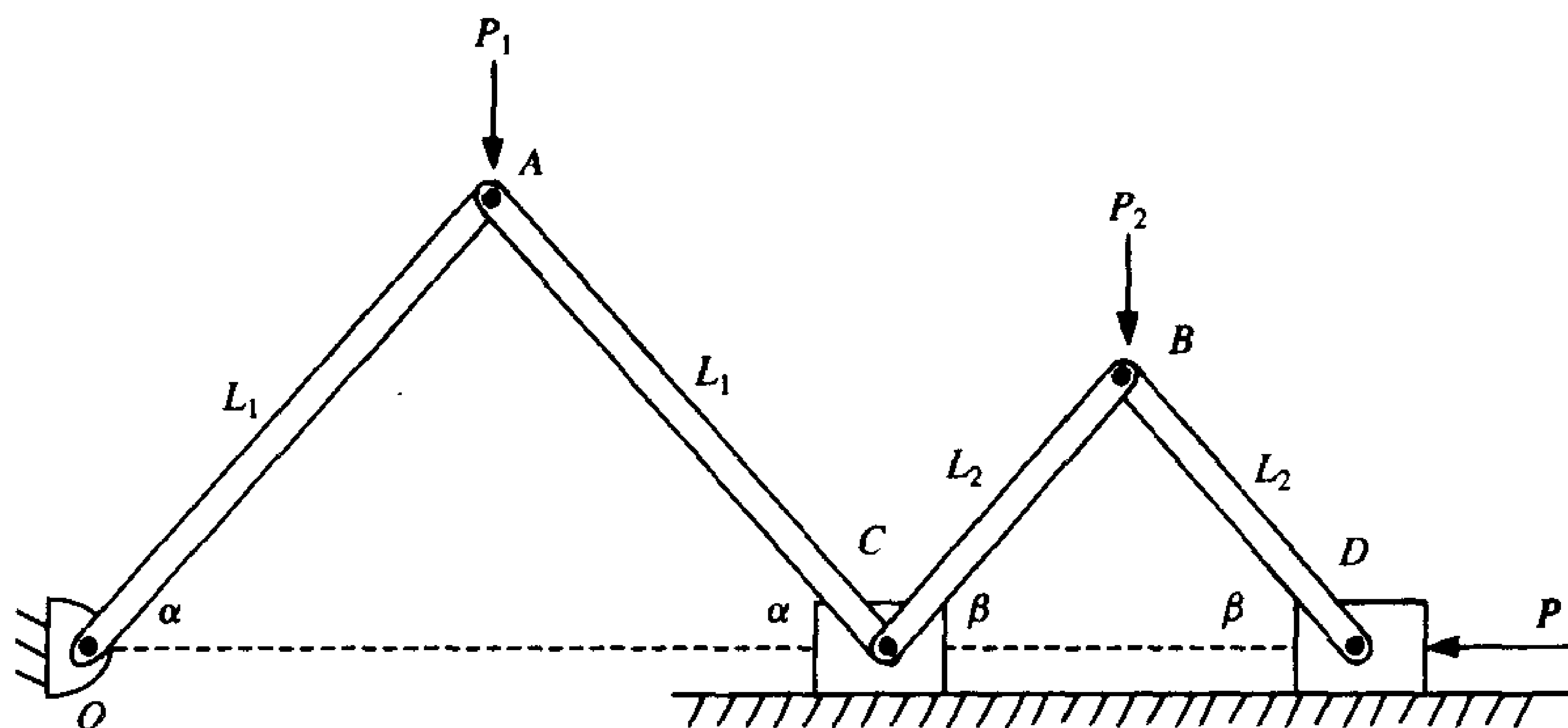


图 13.14　习题 13.1 的两对棒机械装置

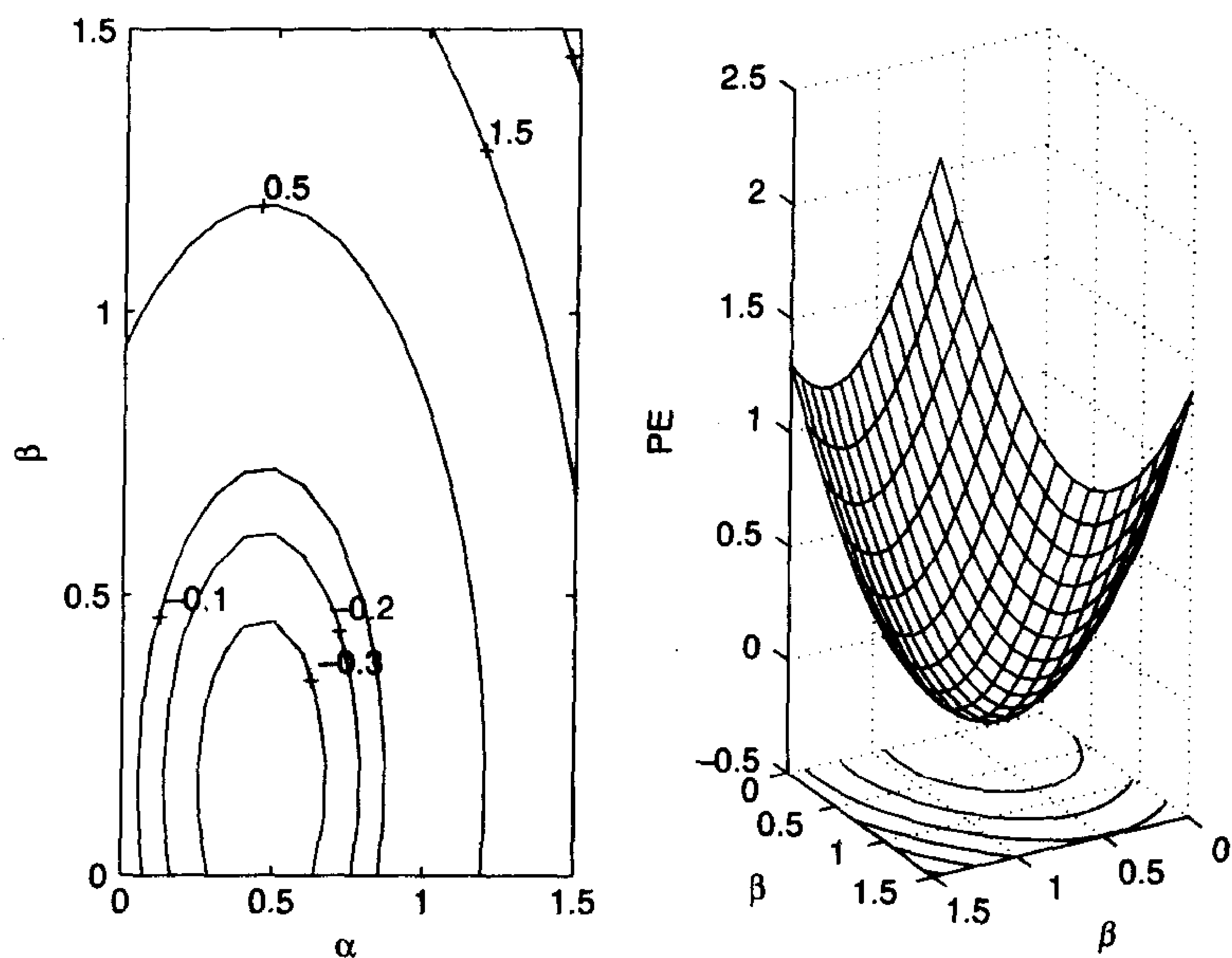


图 13.15　习题 13.1 的等高线图和曲面图

13.2 如图 13.16 所示的管道[①],要求设计一固定横剖面的管道,使其水流速度最大。设计变量为高度 h,底部宽 c,边角 θ。已知流速与润湿周长 p 的倒数成比例关系,而 p 由下式确定:

$$p = c + (2h/\sin\theta)$$

横剖面面积 A 为:

$$A = ch + h^2\cot\theta$$

已知 $A = 100\ \text{ft}^2$,求:

(a) 给出无约束优化问题的定义,以求得流速 $1/p$ 的最大值,其中流速为设计变量的 h 和 θ 函数。

① 参见 P. Y. Papalambros 和 D. J. Wilde, *Principles of Optimal Design: Modeling and Computation*, Cambridge University Press, New York, NY, 1988 年,第 151 页。

(b) 绘出图 13.17。

(c) 用 fminsearch 函数验证由图形得到的解的正确性,给定初始点$(h,\theta)=(1,1\ \text{rad})$。[答案:$[h,\theta]=[7.5938,1.0472\ \text{rad}]$。]

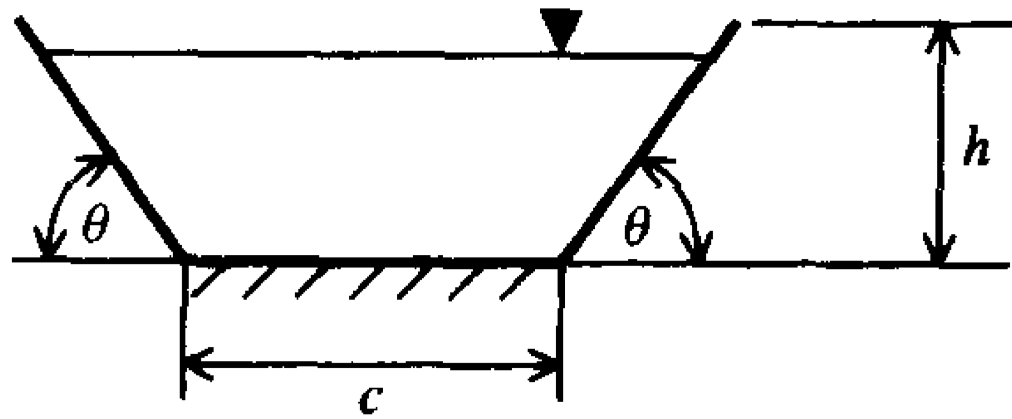


图 13.16 习题 13.2 的水管道问题

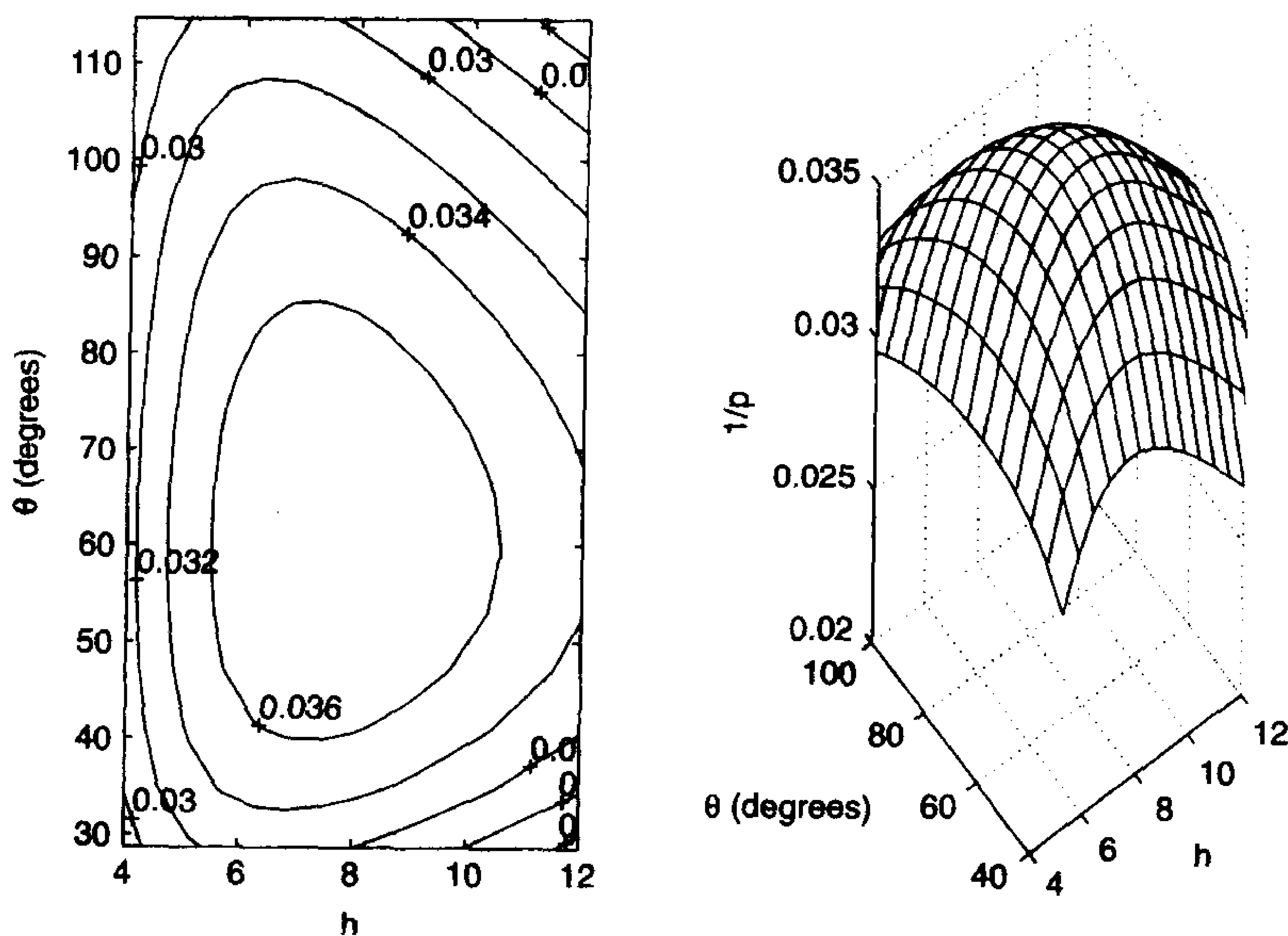


图 13.17 习题 13.2 的等高线图和曲面线图

13.3 图 13.18 是两弹簧系统受力前和受力后的示意图。在 A 点施加一载荷后系统处于变形状态,经过一定时间后将停留在平衡状态 B 点——在该点系统的势能(PE)最小。按以下要求计算 B 点的位移(x_1,x_2):

(a) 绘制如图 13.19 所示的等高线图和曲面图。

(b) 在初始点为$(x_1,\ x_2)\ =\ (1,\ 1)$时用 fminsearch 求最优位移 x_1 和 x_2 的值。[答案:PE = -15.2802, $x_1\ =\ 4.1289$, $x_2=0$。]

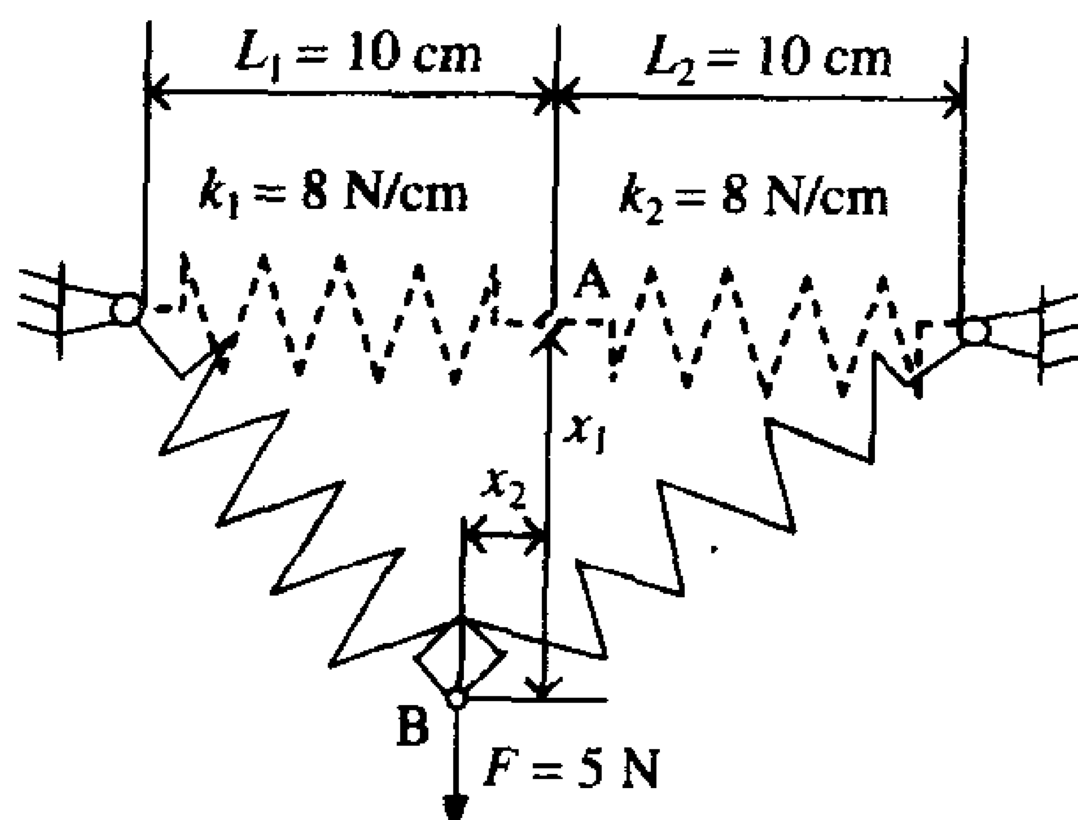


图 13.18 习题 13.4 的两弹簧系统

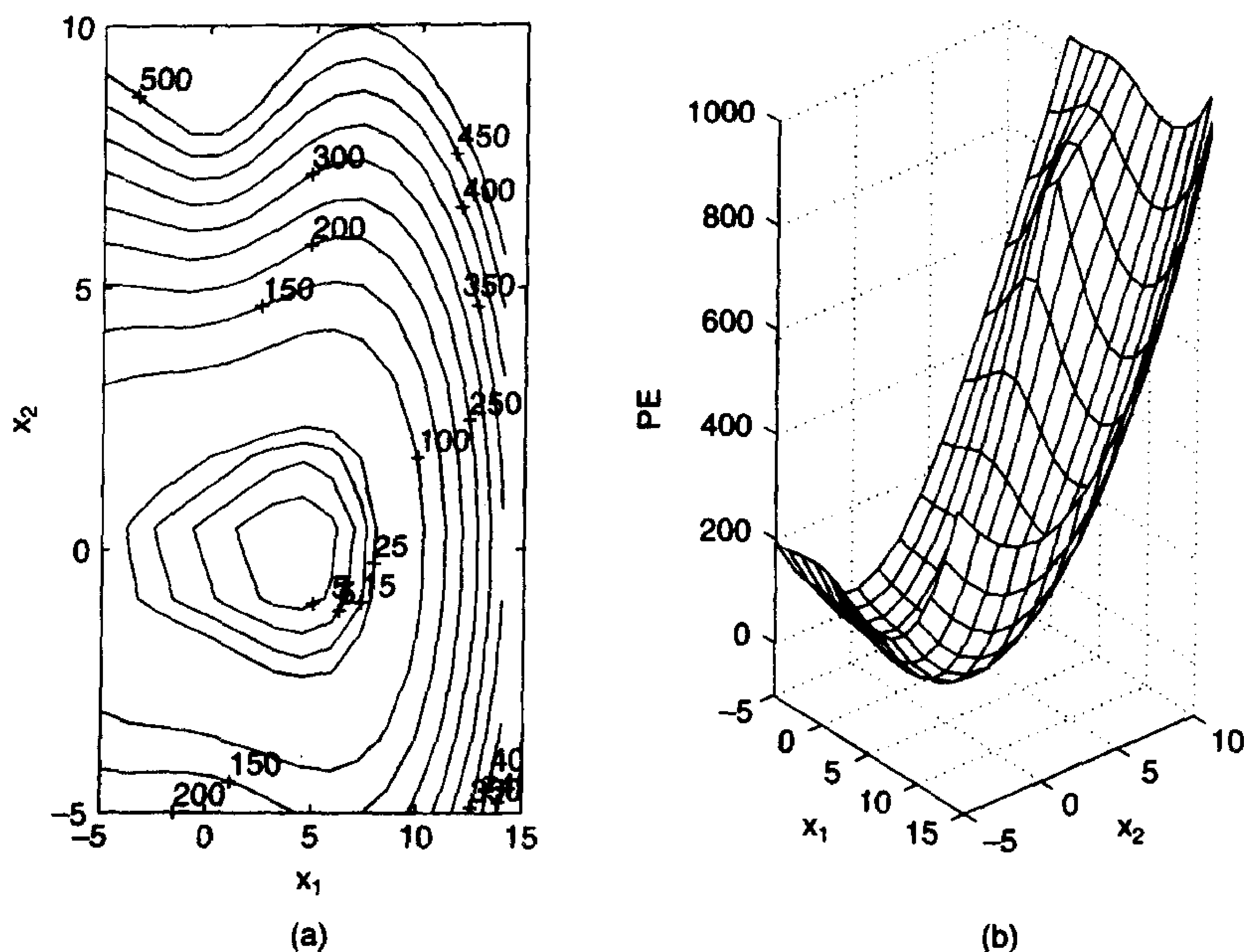


图 13.19 图 13.8 所示的两弹簧系统 PE 函数的图形:(a)等高线图;(b)曲面图

13.4 对一工件进行机械加工所需平均时间由下式确定[①]:

$$T = t_m + \frac{t_m t_c}{T_l} + t_{aux}$$

式中,t_m 为切削时间,T_l 为工具寿命,t_c 为换工具时间,t_{aux}为附加时间。给定经验常数 $t_c = 7, t_{aux} = 3$,在一车削操作中,切削时间由下式确定:

$$t_m = \frac{\pi DL}{1000Vf}$$

式中,已加工面直径为 $D = 100$ mm,需加工的工件长为 $L = 500$ mm,V 和f 分别为表面切削速度和切削工具的进刀速度。切削速度、进刀速度和切削深度 d 对工具寿命的影响程度由泰勒展开式求出,如下所述:

$$VT_1^n f^a d^b = K_t$$

式中,$n = 0.17, a = 0.77, b = 0.37, K_t = 200$, V 和f 为设计变量,则此优化问题可描述如下:

$$\text{minimize} \quad T(V,f)$$
$$\text{s.t.}: f \leqslant 2$$

给定 $d = 0.3$,求:

(a) 绘出其等高线图与曲面图。

(b) 用 fmincon 函数进行数值分析以求得最优解。初始点$(V,f) = (10,1)$。

[答案:$T = 3.942, V = 100.46, f = 2.000$。]

---

① 参见 D.A.Stephenson 和 J.S.Agapiou, *Metal Cutting Theory and Practice*, Marcel Dekker, New York, NY, 1997 年。

13.5 图13.20中两个无摩擦拖车A和B由三个弹簧相连,弹簧弹力与其伸长量之间为线性关系,弹簧的倔强系数分别为 $k_1=5$ N/m, $k_2=10$ N/m 和 $k_3=8$ N/m[①]。当外力 P 为0时,弹簧处于自然平衡位置。给出热能函数如下。

$$\text{minimize} \quad \text{PE} = 0.5k_2x_1^2 + 0.5k_3(x_2-x_1)^2 + 0.5k_1x_2^2 - Px_2$$

(a) 试画出等高线图和曲面图,如图13.21所示。

(b) 用 fminunc 计算初始点 $(x_1, x_2) = (1,1)$, $P = 100$ N 时,位移 x_1 与 x_2 的解。

[答案:PE = 529.41, $x_1 = 4.759$, $x_2 = 10.5882$。]

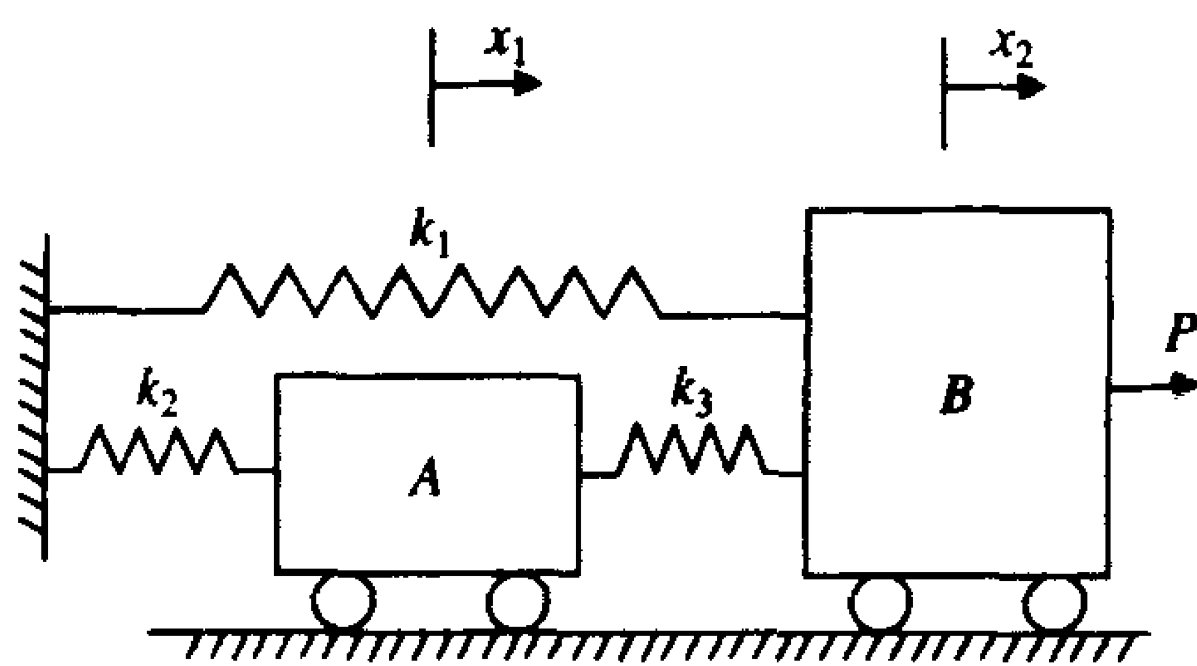


图13.20 习题13.5中的弹簧-小车

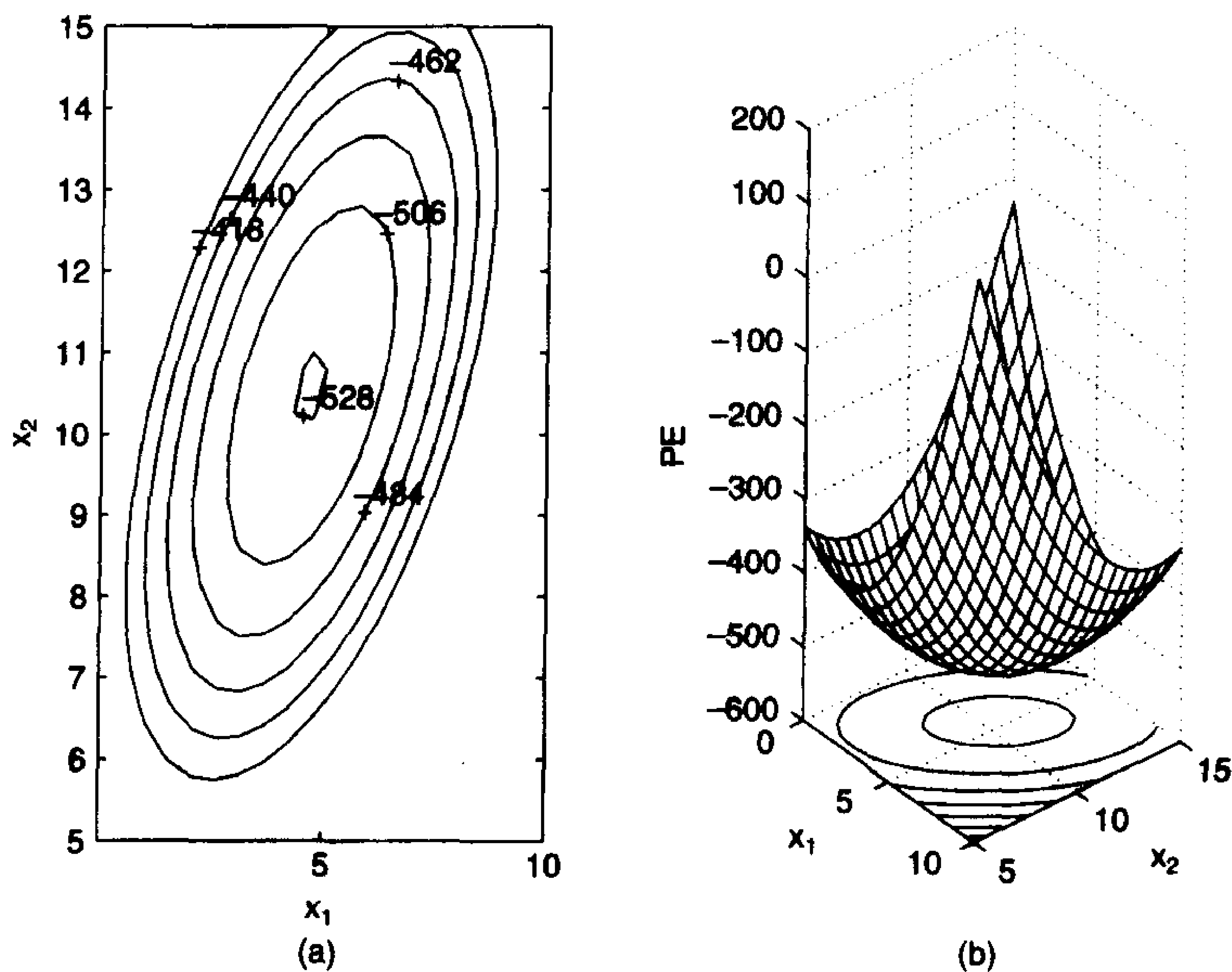


图13.21 图13.20所示的两弹簧系统PE函数的图形:(a) 等高线图;(b) 曲面图

13.6 如图13.22所示[②],有三个小车(重物)用弹簧连接,开始处于未受力平衡状态。当分别施加外力 P_1, P_2 和 P_3 时,小车状态将改变。三小车从其初始平衡位置($x_i=0$, i 值取1,2,3)处开始振动,其位移可通过求系统的势能(PE)的极小值求得:

$$\text{PE} = 0.5X'KX - X'P$$

---

① 参见S.S.Rao, *Engineering Optimization, Theory and Practice*, 第三版, John Wiley & Sons, New York, NY, 1996年。

② 参见S.S Rao,出处同上,第420页。

其中,

$$K=\begin{bmatrix} k_1+k_2+k_4 & -k_3 & -k_4 \\ -k_3 & k_2+k_3+k_5 & -k_5 \\ -k_4 & -k_5 & k_4+k_5+k_6 \end{bmatrix}$$

$$P=[P_1 \quad P_2 \quad P_3]'$$

$$X=[x_1 \quad x_2 \quad x_3]'$$

输入数据为:

$k_1=4500$ N/m	$k_4=2250$ N/m	$P_1=1100$ N
$k_2=1650$ N/m	$k_5=550$ N/m	$P_2=1800$ N
$k_3=1100$ N/m	$k_6=9300$ N/m	$P_3=3300$ N

试用 fminunc 函数求出三小车的平衡位置。初始点$(x_1,x_2,x_3)=(0,0,0)$。

[答案:$[x_1,x_2,x_3]=[0.322,0.714,0.365]$。]

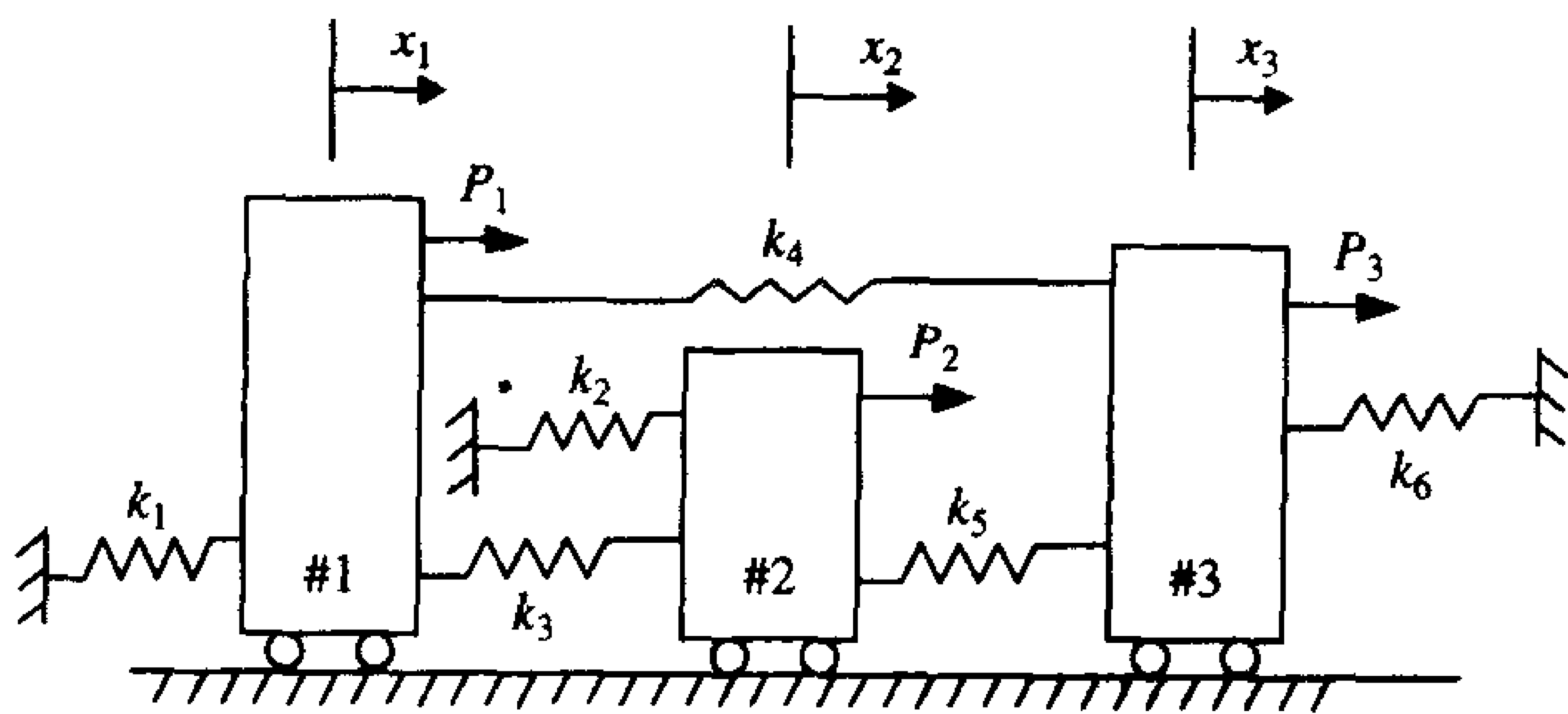


图 13.22 习题 13.6 的弹簧-小车系统

13.7 如图 13.23 所示的弹簧-重物系统,未加重物时弹簧处于不变形的位置,当在每段弹簧之间挂上重物时弹簧将变形①。第 i 根弹簧的稳定性用 k_i 表示,由下式得出:

$$k_i=450+225(N/3-i)^2 \quad \text{N/m} \qquad i=1,\cdots,6$$

其中,$N=5$ 是所挂重物的个数,W_j 由下式确定:

$$W_j=60j \quad \text{N} \quad j=1,\cdots,5$$

挂重物前每段弹簧长为 $L_i=7.5$ m,弹簧连接点(从 2 到 6)的坐标用 10 个设计变量:(X_i,Y_i),$i=2,\cdots,6$ 表示。可通过求以下 PE 函数的极小值来获得其平衡位置:

$$\text{PE}=0.5\sum_{i=1}^{6}K_i\Delta L_i^2+\sum_{j=1}^{5}W_jY_{j+1}$$

其中,

$$\Delta L_i=\sqrt{(X_{i+1}-X_i)^2+(Y_{i+1}-Y_i)^2}-L_i$$

要求用 fminunc 求出平衡位置,即图 13.23 中弹簧变形后的各连接点坐标。初始点$(X_2,X_3,X_4,X_5,X_6,Y_2,Y_3,Y_4,Y_5,Y_6)=(7.5,15,22.5,30,37.5,0,0,0,0,0)$。

[答案: $X_2=7.9168$, $X_3=16.2538$, $X_4=24.3557$, $X_5=32.223$, $X_6=39.303$, $Y_2=$

① 参见 G.N. Vanderplaats,出处同上,第 94 页。

$-3.8577, Y_3 = -7.1458, Y_4 = -8.8364, Y_5 = -8.2859, Y_6 = -5.1603$。]

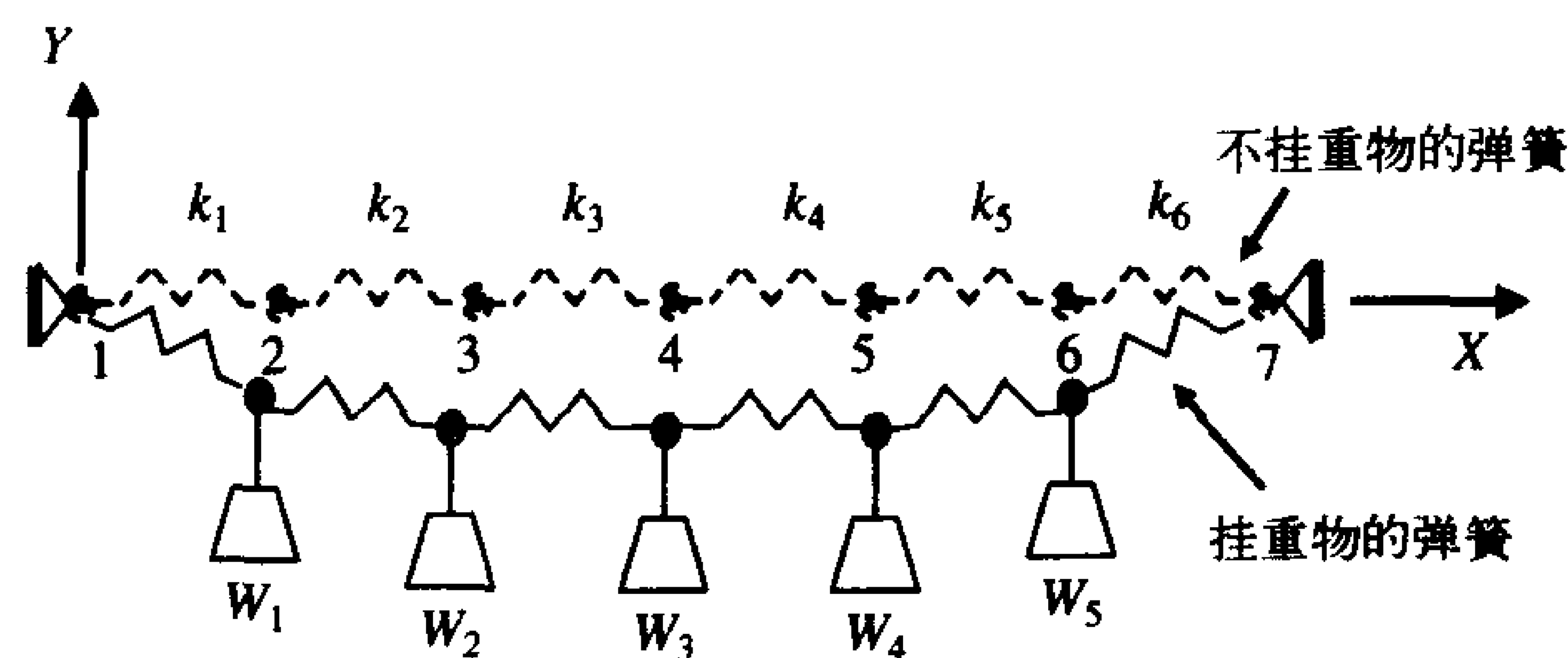


图 13.23 习题 13.6 中的弹簧 - 重物系统的初始位置和变形后的位置

13.8 如图 13.24 所示,已知管柱的弯曲临界载荷 F 符合下述方程式,其中 a,b 为未知常数:

$$F = \frac{\pi^a E R^b t^{L_i - b}}{4L^2}$$

式中,E 为弹性模量,R 为平均半径,t 为厚度,L 为柱长。假定导致弯曲的精确关系式未知,常数将由实验数据经曲线拟合得到。为此进行了实验,实验所用柱体半径不同,弹性模量 $E = 250$ ksi,长度 $L = 5$ in,厚度 $t = 1$ in,给柱施加屈曲载荷。表 13.7列出了不同 R 值对应的屈曲载荷 P。要求用 `lsqcurvefit` 函数求 a 和 b,初始点$(a,b) = (1,1)$,重新画出图 13.25,并在图中给出实验数据和曲线拟合数据。拟合曲线效果是否很好?为什么好或不好?

[答案:$a = 2.7163, b = 3.1599$。]

表 13.7 习题 13.8 中的半径 R 和载荷 P 的实验数据

实验次数	R(in)	F(klb)
1	1.1	86.6
2	1.7	120.5
3	2.1	520.88
4	2.9	1758
5	3.9	4098

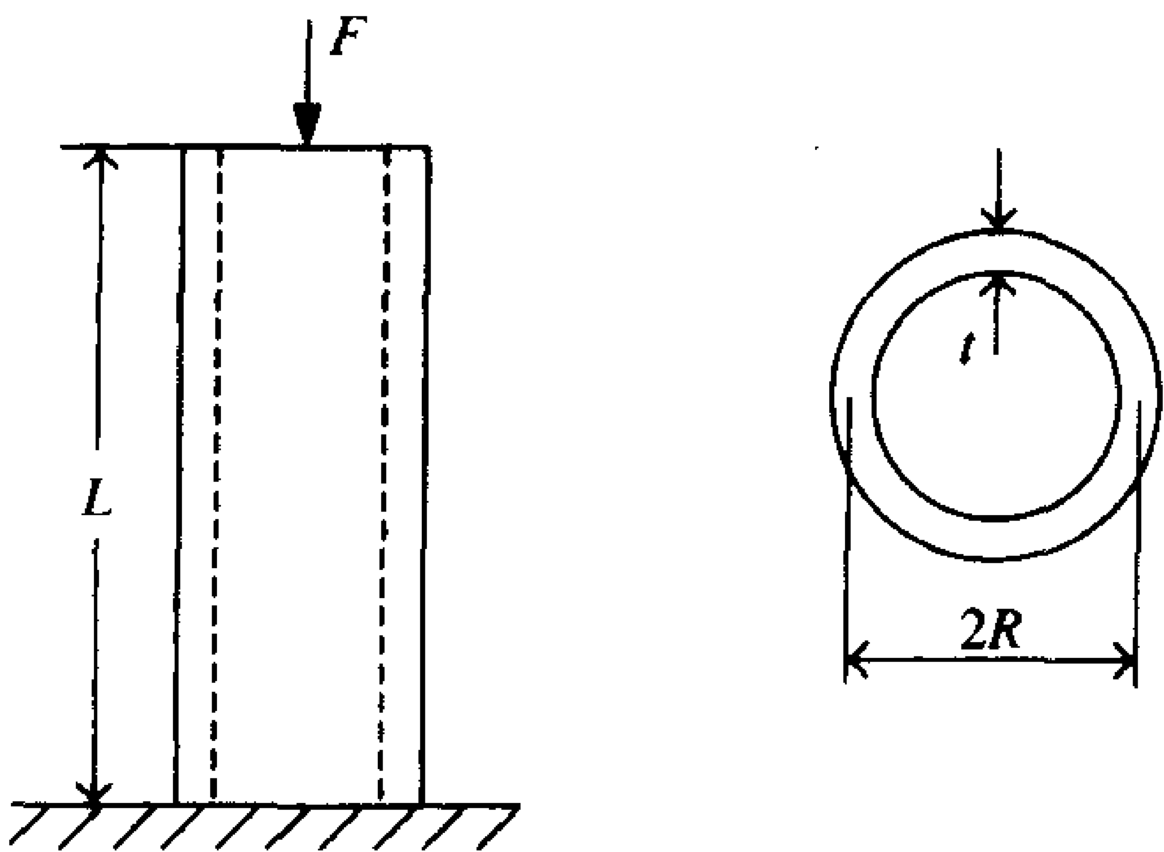


图 13.24 习题 13.8 中的柱体几何形状

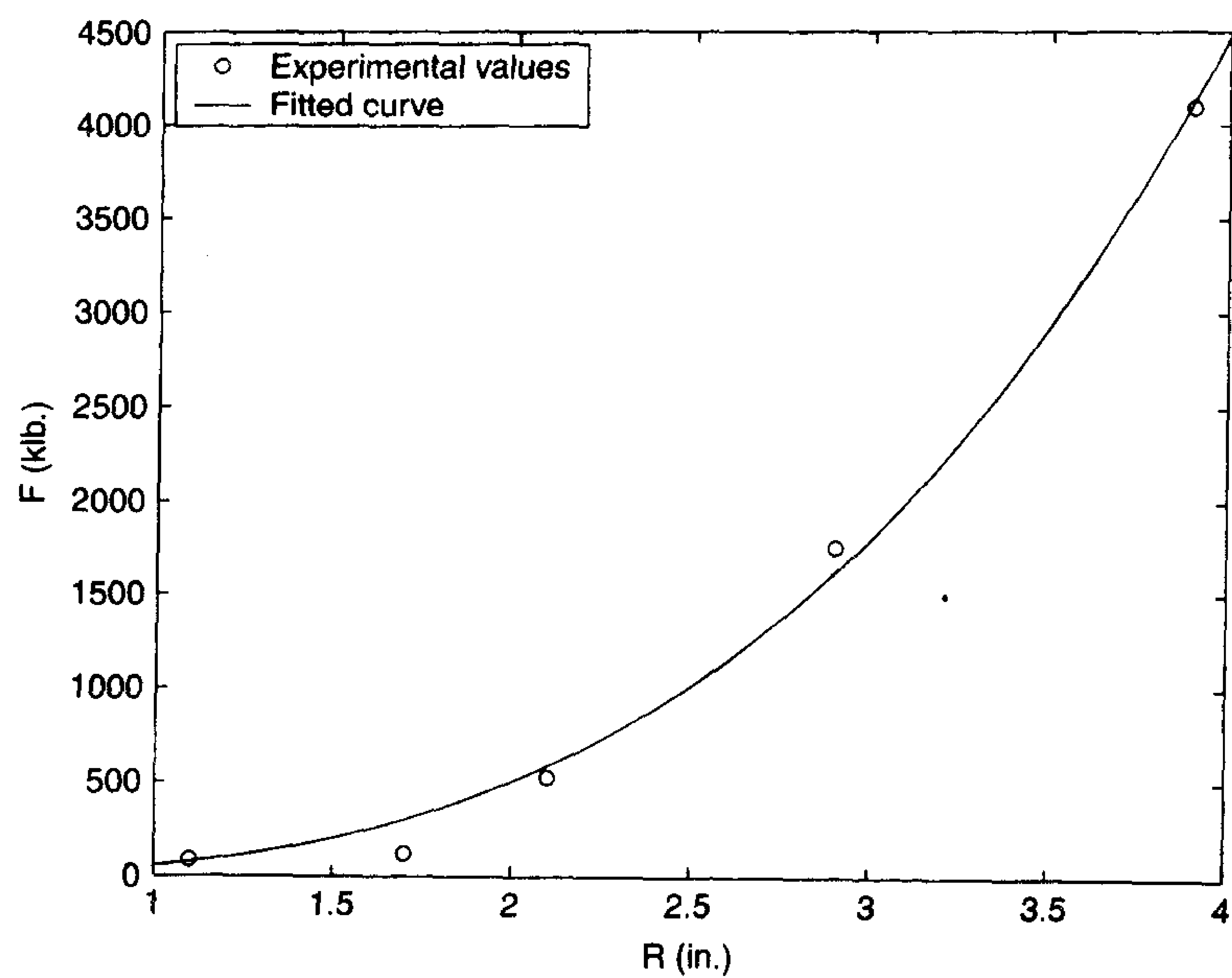


图 13.25　习题 13.8 中管状柱体问题的结果曲线

13.9　假设一种药品中某一成分的含量应保持在药品总量的 0.5%,而这种成分随时间的推移会减少。药品上市前八个星期内,存在着下降到 0.5% 的可能,另外还会有其他不可控因素的出现,所以用理论方法预测其今后的下降趋势并不可行。要辅助管理者做出是否对库存过期药品废弃或更换的决定,应定期分析药品以测量药品成分的含量。表 13.8给出其测量值,用下述的非线性模型可说明表中数据的变化规律。

$$Y = a + (0.51 - a)e^{-b(t-8)}$$

现要求用`lsqcurvefit`函数估算出非线性模型中的参数 a 和 b 值。初始点$(a,b)=(1,1)$。拟合模型好不好?

[答案:$[a,b]=[0.3918,0.1394]$。]

表 13.8　习题 13.9 的数据表

自生产之日起的时间长度(周数)t	成分含量 Y	自生产之日起的时间长度(周数)t	成分含量 Y
7	0.488	25	0.405
9	0.473	27	0.403
11	0.448	29	0.391
13	0.435	31	0.403
15	0.431	33	0.398
17	0.453	35	0.393
19	0.421	37	0.398
21	0.405	39	0.388
23	0.405	41	0.388

13.10 观察停车处的车辆,车停下来然后又加速行驶越过五座房屋。假设这五座房屋与停车处的距离已知,从车开始加速行驶时刻算起,到驶过每座房屋时记下其行驶时间,表 13.9 中列出了这些数据。车辆加速行驶问题是时间的函数,其形式为:

$$a(t) = Ct^2 + Dt + a_0$$

式中,t 为行驶时间(s),a_0 为初始加速度(ft/s²),C 和 D 为常数,则车行驶的位移随时间的变化关系由下式确定:

$$x(t) = At^4 + Bt^3 + a_0 t^2$$

试用 lsqnonlin 函数估计出车速 $v(t)$随时间变化的函数关系。初始点$(A,B) = (0,0)$,已知车的初始位移、初始速度、初始加速度分别为 $x_0 = 0$ ft, $v_0 = 0$ ft/s, $a_0 = 2$ ft/s²,计算最小二乘法产生的误差。如果误差很大,试给出一种方法减小该误差。

[答案:$v(t) = -0.1084t^3 + 0.8307t^2 + 4t$。]

表 13.9　习题 13.10 中车辆的位移与行驶时间

位置(ft)	时间(s)
0	0
9	2.05
20	3.1
60	4.8
90	5.6
120	6.8

13.11 在一个球体表面有四个点:$A = (1, 1, 1)$, $B = (-1, 2, 4)$, $C = (2, 3, 4)$, $D = (-3, -4, 1)$,球体包含在球体表面内部,括号内的数分别是点的 x,y 和 z 坐标值。求当球具有最小直径时的球心坐标及其最小半径。将球体中心坐标(x_c, y_c, z_c)及半径 R 设为变量。用 fmincon 求解,初始点为$(x_c, y_c, z_c, R) = (4, 4, 4, 4)$。

[答案:$[x, y, z, R] = [-1, -1.5, -1, 3.2016]$。]

13.12 如图 13.26 所示,一个三杆桁架,以施力点的垂直偏移作为目标函数①,如下所述:

$$\text{minimize } f = \frac{Ph}{E}\frac{1}{x_1 + \sqrt{2}x_2}$$

其中杆的横截面积为 $A_1 = x_1, A_2 = x_2$,设计变量为 x_1 和 x_2,所施加力 P 的方向如图13.26 中所示。

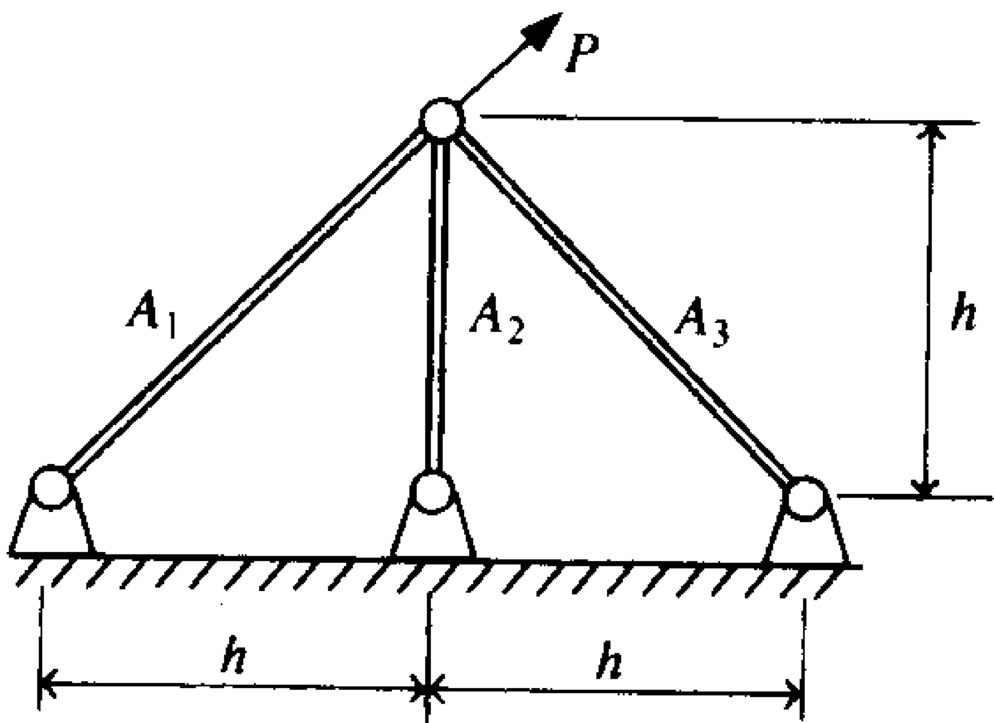


图 13.26　练习 13.12 中的三杆桁架

① 参见 S.S.Rao,出处同上,第 530 页。

约束条件包括三杆实际应力的约束及设计变量的上下限约束,如下所示:

$$P\frac{x_2+\sqrt{2}x_1}{\sqrt{2}x_1^2+2x_1x_2}-\sigma^{(u)}\leqslant 0$$

$$P\frac{1}{x_1+\sqrt{2}x_2}-\sigma^{(u)}\leqslant 0$$

$$P\frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}+\sigma^{(l)}\leqslant 0$$

$$x_i^{(l)}\leqslant x_i\leqslant x_i^{(u)}\quad i=1,2$$

式中,$\sigma^{(u)}$是张应力的最大值,$\sigma^{(l)}$为压缩应力的最大值。$x_i^{(l)}$ 为 x_i 的下限,$x_i^{(u)}$ 为 x_i 的上限。上述各参数值为 $\sigma^{(u)}=17.5,\sigma^{(l)}=-12,x_i^{(l)}=0.2,x_i^{(u)}=6.0(i=1,2)$,$P=25,E=2,h=2$。用 fmincon 函数求出横截面及垂直偏移的最优解。已知初始条件$(x_1,x_2)=(0,0)$。

[答案:$[x_1,x_2]=[6.00,6.00]$和$f=1.7259$。]

13.13 如图 13.27 所示的电桥网络,有 5 个电阻 R_i,流过每个电阻的电流为 $I_i,i=1,\cdots,5$,电阻上的电压压降为 $V_i=R_iI_i$。假定 R_1 上的压降为 $V_1=3$ V,R_3 上的压降为 $V_3=1$ V,R_5 上的压降为 $V_5=1$ V,且流过所有电阻的电流在下限 1 安培和上限 2 安培间变化。

(a)用公式描述一约束优化问题,从而求出总功率最小时的 R_i 值。

(b)用 fmincon 函数求出每个电阻的最优解及总功率值。已知 R_i 上的功率等于 $I_i^2R_i$,由基尔霍夫电流第一定律可知流入节点(图 13.27 中的 A 点或 B 点)的所有支路电流的代数和为 0。

[答案:$[R_1,R_2,R_3,R_4,R_5]=[3,1,1,2,0.5]$欧姆,总功率 = 12 W。]

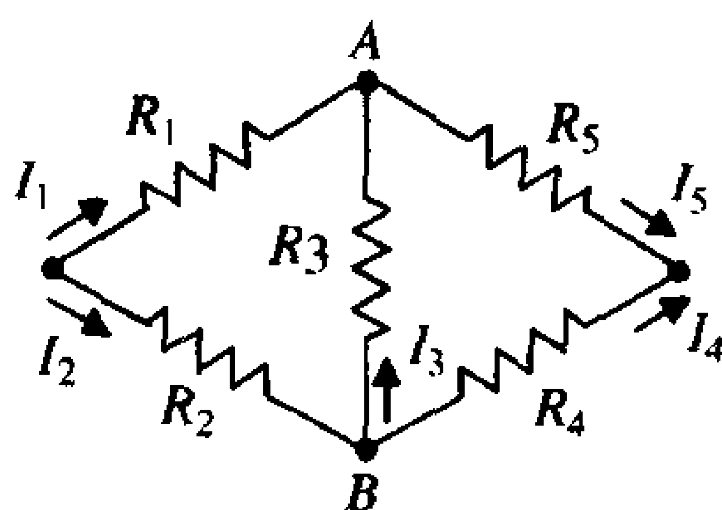


图 13.27 习题 13.13 的电桥网络

13.14 一公司有 m 家工厂用于生产某产品。该产品经轮船托运到 n 个仓库存放,第 j 处的仓库至少存 b_j 件产品,第 i 处的生产工厂能生产 a_i 件产品。从第 i 家工厂托运到仓库j。x_{ij}件产品的运费为$c_{ij}x_{ij}+d_{ij}x_{ij}^2$,式中 c_{ij}和 d_{ij}均为常数。上述问题可描述为二次规化问题,如下所示:

$$\text{minimize}\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij}+d_{ij}x_{ij}^2)$$

$$\text{s.t.:}\sum_{i=1}^{m}x_{ij}\geqslant b_j\qquad j=1,\cdots,n$$

$$\sum_{j=1}^{n} x_{ij} \leqslant a_i \qquad i = 1, \cdots, m$$

$$x_{ij} \geqslant 0 \qquad \text{对所有 } i,j \text{ 而言}$$

假定 $m=6, n=4, \boldsymbol{a}=[8,24,20,24,16,12]', \boldsymbol{b}=[29,41,13,21]'$，则：

$$\boldsymbol{c} = \begin{bmatrix} 300 & 270 & 460 & 800 \\ 740 & 600 & 540 & 380 \\ 300 & 490 & 380 & 760 \\ 430 & 250 & 390 & 600 \\ 210 & 830 & 470 & 680 \\ 360 & 290 & 400 & 310 \end{bmatrix}$$

$$\boldsymbol{d} = (-1)\begin{bmatrix} 7 & 4 & 6 & 8 \\ 12 & 9 & 14 & 7 \\ 13 & 12 & 8 & 4 \\ 7 & 9 & 16 & 8 \\ 4 & 10 & 21 & 13 \\ 17 & 9 & 8 & 4 \end{bmatrix}$$

试用 quadprog 函数求出由第 i 家工厂生产并托运到第 j 个仓库的产品数的最优解 x_{ij}。

[答案：$[x_{11}, x_{21}, x_{31}, x_{41}, x_{51}, x_{61}; x_{12}, x_{22}, x_{32}, x_{42}, x_{52}, x_{62}; x_{13}, x_{23}, x_{33}, x_{43}, x_{53}, x_{63}; x_{14}, x_{24}, x_{34}, x_{44}, x_{54}, x_{64}]' = [0,0,20,0,3,6,8,3,0,24,0,6,0,0,0,0,13,0,0,21,0,0,0,0]'$。]

13.15 如图 13.28 所示一个两杆桁架关于 y 轴对称。杆 1 和杆 2 的位置用无量纲值 x/h 表示，杆的横截面面积用无量纲值 A/A_{ref} 表示。此两项分别作为设计变量 x_1 和 x_2，其中 A_{ref} 为 A 的参考值，h 为桁架高度，所施力 P 的方向在 $-90° \leqslant \theta \leqslant 90°$ 间变化。要求计算出桁架重量 w 的最小值：

$$-\sigma_0 \leqslant \frac{P\sqrt{1+x_1^2}(x_1\cos\theta + \sin\theta)}{2x_1 x_2 A_{ref}} \leqslant \sigma_0$$

$$-\sigma_0 \leqslant \frac{P\sqrt{1+x_1^2}(x_1\cos\theta - \sin\theta)}{2x_1 x_2 A_{ref}} \leqslant \sigma_0$$

$$w = 2phx_2 A_{ref}\sqrt{1+x_1^2}$$

式中，ρ 为密度，约束条件包括杆 1 和杆 2 上的应力约束。其中，P 为所施加的外力。此外，设计变量 x_1, x_2 的下限和上限为：

$$x_i^{\min} \leqslant x_i \leqslant x_i^{\max} \qquad i = 1,2$$

表 13.10 中列出各参数的相应值，试用 fseminf 函数求出设计变量的最优解并画出应力的曲线图。要求该应力为 θ 的函数，优化过程中采样间隔取 5°。若给定初始值：

$$(x_1^0, x_2^0) = (0.1, 0.1)$$

结果如图 13.29 所示。

[答案：$[x_1, x_2] = [0.8025, 0.4517]$，$w = 27.8766$。]

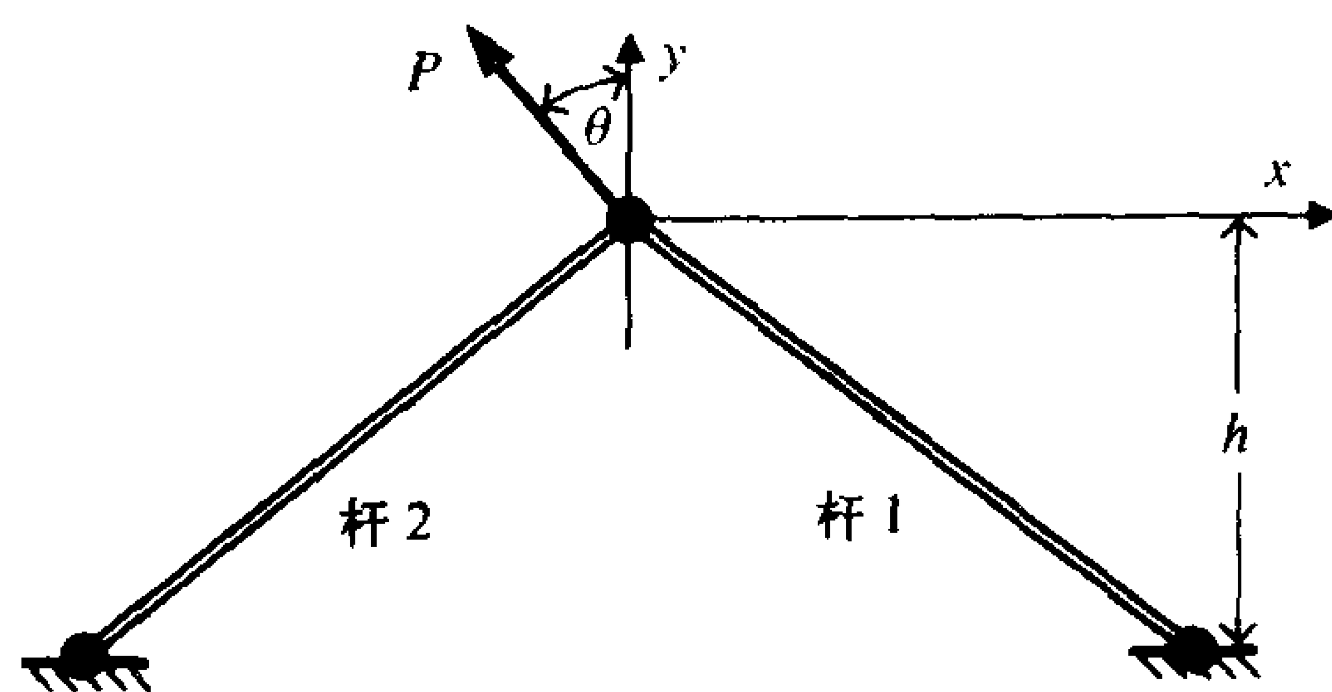


图 13.28　习题 13.15 的两杆桁架

表 13.10　习题 13.15 中的参数输入值

ρ(lb/in³)	P(lb)	σ_0(psi)	h(in)	A_{ref}(in²)	x_1^{min}	x_2^{min}	x_1^{max}	x_2^{max}
0.283	8000	18 500	85	1	0.15	0.15	3.0	3.5

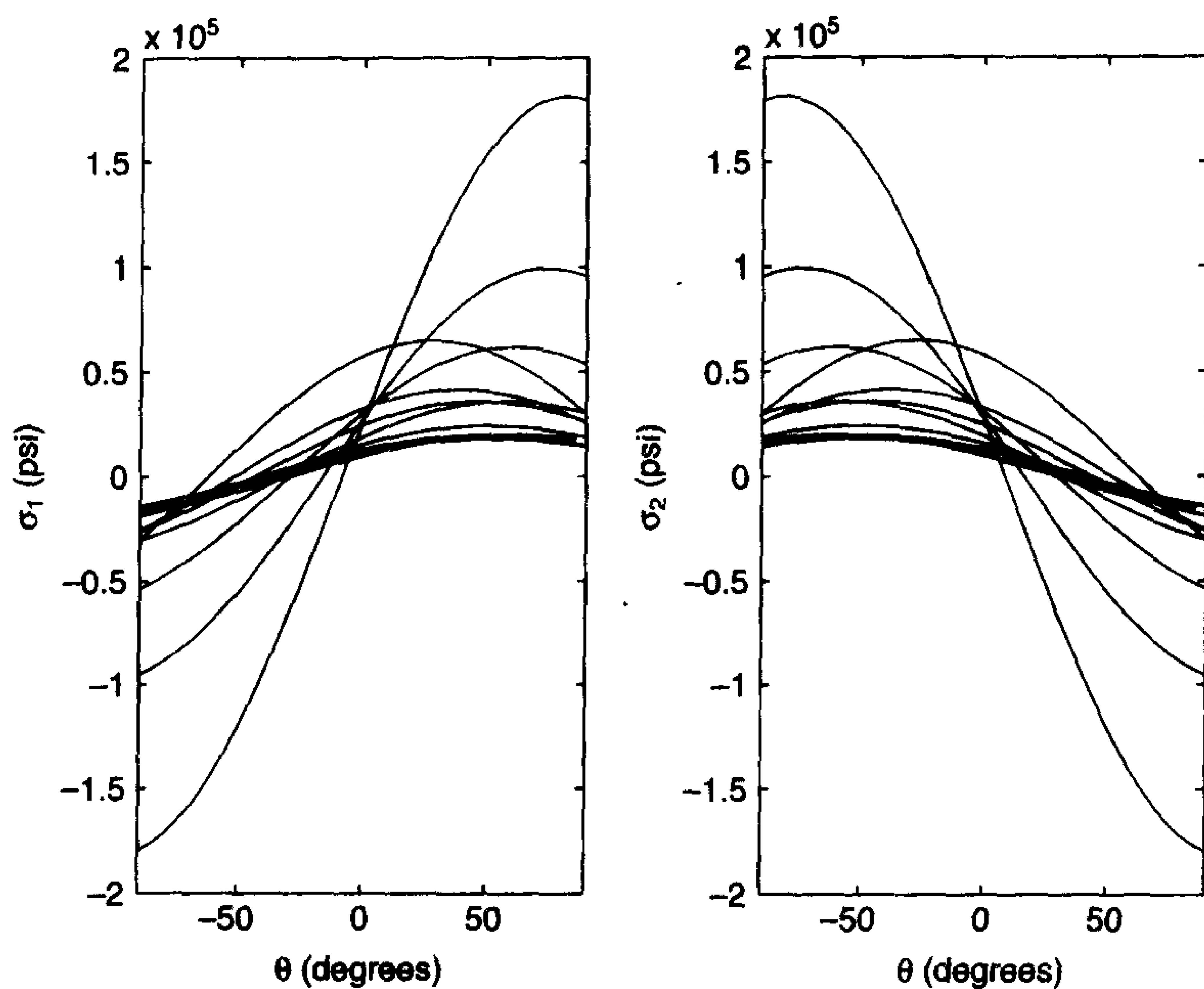


图 13.29　由优化过程得到的图 13.28 中的杆 1 和杆 2 的应力曲线图

参考文献

1. J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, NY, 1989.
2. M. Austin and D. Chancogne, *Engineering Programming in C, MATLAB and JAVA*, John Wiley & Sons, New York, NY, 1998.
3. M. Branch and A. Grace, *MATLAB Optimization Toolbox User's Guide*, The Math Works, Natick, MA, 1996.
4. V. Changkong and Y. Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*, Elsevier Science Publishing Co., New York, NY, 1983.
5. N. Draper and H. Smith, *Applied Regression Analysis*, John Wiley & Sons, New York, NY, 1966.
6. H. Eschenauer, J. Koski, and A. Osyczka, Eds., *Multicriteria Design Optimization*, Springer-

Verlag, New York, NY, 1990.

7. J. Golinski, "Optimum Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods," *Journal of Mechanisms*, Vol. 5, 1970, pp. 287–309.
8. C. Gosselin and J. Angeles, "A Global Performance Index for the Kinetic Optimization of Robotic Manipulators," *ASME Journal of Mechanical Design*, Vol. 113, September 1991, p. 222.
9. U. Kirsch, *Optimal Structural Design*, McGraw-Hill, New York, NY, 1981.
10. A. Messac, "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal* 34, 1996, pp. 149–158.
11. A. Osyczka, *Multicriterion Optimization in Engineering with Fortran Programs*, Ellis Horwood Limited, West Sussex, England, 1984.
12. P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design*, Cambridge University Press, Cambridge, England, 1988.
13. S. S. Rao, *Engineering Optimization, Theory, and Practice*, 3rd ed., John Wiley & Sons, New York, NY, 1996.
14. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization*, John Wiley & Sons, New York, NY, 1983.
15. J. Shigley and C. Mischke, *Mechanical Engineering Design*, McGraw-Hill, New York, NY, 1989.
16. D. A. Stephenson and J. S. Agapiou, *Metal Cutting Theory and Practice*, Marcel Dekker, New York, NY, 1997.
17. G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, NY, 1984.
18. D. A. Van Veldhuizen, and G. B. Lamont, "Multi-Objective Evolutionary Algorithm Research: A History and Analysis," Technical Report TR-98-03, Air Force Institute of Technology, Wright Patterson AFB, OH, 1998.
19. D. J. Wilde, *Globally Optimal Design*, John Wiley & Sons, New York, NY, 1978.

第14章 工程统计

本章介绍了 Statistics 工具箱在许多工程统计问题方面的应用。

14.1 统计量描述

有一组测量值 $x_j, j=1,2,\cdots,n$,其样本平均值为:

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \qquad (14.1)$$

样本方差为:

$$s^2 = \frac{1}{n-1}\left[\sum_{j=1}^{n} x_j^2 - n\bar{x}^2\right] \qquad (14.2)$$

其中,s 为标准偏差。这里的 $\bar{x}$ 和 s 分别为平均值 μ 和标准偏差 σ 的估计值。求平均值的方法如下:

```
mean(x)
```

求标准偏差的方法如下:

```
std(x)
```

其中,$\boldsymbol{x}$ 为测量值向量或测量值矩阵。

将测量值分布的区域分为相等的 N 段,每段称为一区间。该区间内的 x_j 应大于等于其下限值,小于上限值,每个区间的中心元素用 $b_k, k=1,2,\cdots,N$ 表示。当所有 x_j 均分配到相应区间后,即可计算出该区间内的总数,用 n_k 表示,n_k 即为分布于中心元素为 b_k 的区间内数据的个数。如果将 n_k 看做其中心元素的一个函数,每段区间用一柱条表示,其宽度为区间上限与下限之差,这样得到的图称为直方图。每段区间内所含数据的个数由以下语句求得:

```
[nk,b] = hist(x,N)
```

其中,**nk** 为 n_k 的向量,**b** 为 hist 函数求出的区间中心值向量,x 为 n 个采样数据,N 为所分区间数。如果 N 值默认,则 MATLAB 将采用 $N=10$。不带返回值的 hist 函数用于画出直方图,其用法如下:

```
hist(x)
```

但 hist 函数常用于求 n_k。还可以用

```
bar(b,nk)
```

语句得到直方图。

定义 $f_k = n_k/n$,可得到分布于中心元素为 b_k 的区间内元素数与样本容量 n 的比值。令:

$$c_k = \sum_{j=1}^{k} f_j \qquad k = 1,2,\cdots,N$$

则称 c_k 为累积分布函数,可由以下函数求得:

```
ck = cumsum(f)
```

其中,$f=[f_1\ f_2\cdots f_k]=[n_1/n\ \ n_2/n\cdots n_k/n]$。还可以画出 c_k 对 b_k 的变化图,该图近似描述了一测量值小于或等于 b_k 的概率。

将 x 按从小到大的顺序排序,最小值可用 min(x)求得,最大值由 max(x)求得,其中 $x=[x_1,x_2,\cdots,x_n]$区间范围等于样本的最大值与最小值之差,可由语句:

```
range(x)
```

或

```
max(x) - min(x)
```

求得。排序值的中心元素称为中位数。如果样本容量 n 为奇数,则中位数等于 $x_{(n+1)/2}$;如果为偶数,则中位数等于$(x_{n/2}+x_{n/2+1})/2$。中位数可由以下函数求得:

```
median(x)
```

另一种常用的统计量为几何均值。几何均值定义为样本中 n 个测量数据乘积的 n 次方根,即:

$$\bar{x}_g=\sqrt[n]{\prod_{j=1}^{n}x_j}$$

求几何均值的语句为:

```
geomean(x)
```

或

```
prod(x)^(1/length(x))
```

下面举例说明这些关系式的使用。

例 14.1　求几个统计量

对表 14.1 中的数据求平均值、中位数、标准方差、几何均值、区间范围、最小值、最大值,然后画出直方图及累积分布函数曲线。数据分放在 9 个区间,区间起始值为 80,终止值为 240。图中表示区间的每个柱条为黄色。将表中的数据放于函数 **DataSet141** 中,其定义为:

```
function d = DataSet141
d = [105 97 245 163 207 134 218 199...
   160 196 221 154 228 131 180 178...
   157 151 175 201 183 153 174 154...
   190 76 101 142 149 200 186 174...
   199 115 193 167 171 163 87 176...
   121 120 181 160 194 184 165 145...
   160 150 181 168 158 208 133 135...
   172 171 237 170 180 167 176 158...
   156 229 158 148 150 118 143 141...
   110 133 123 146 169 158 135 149];
```

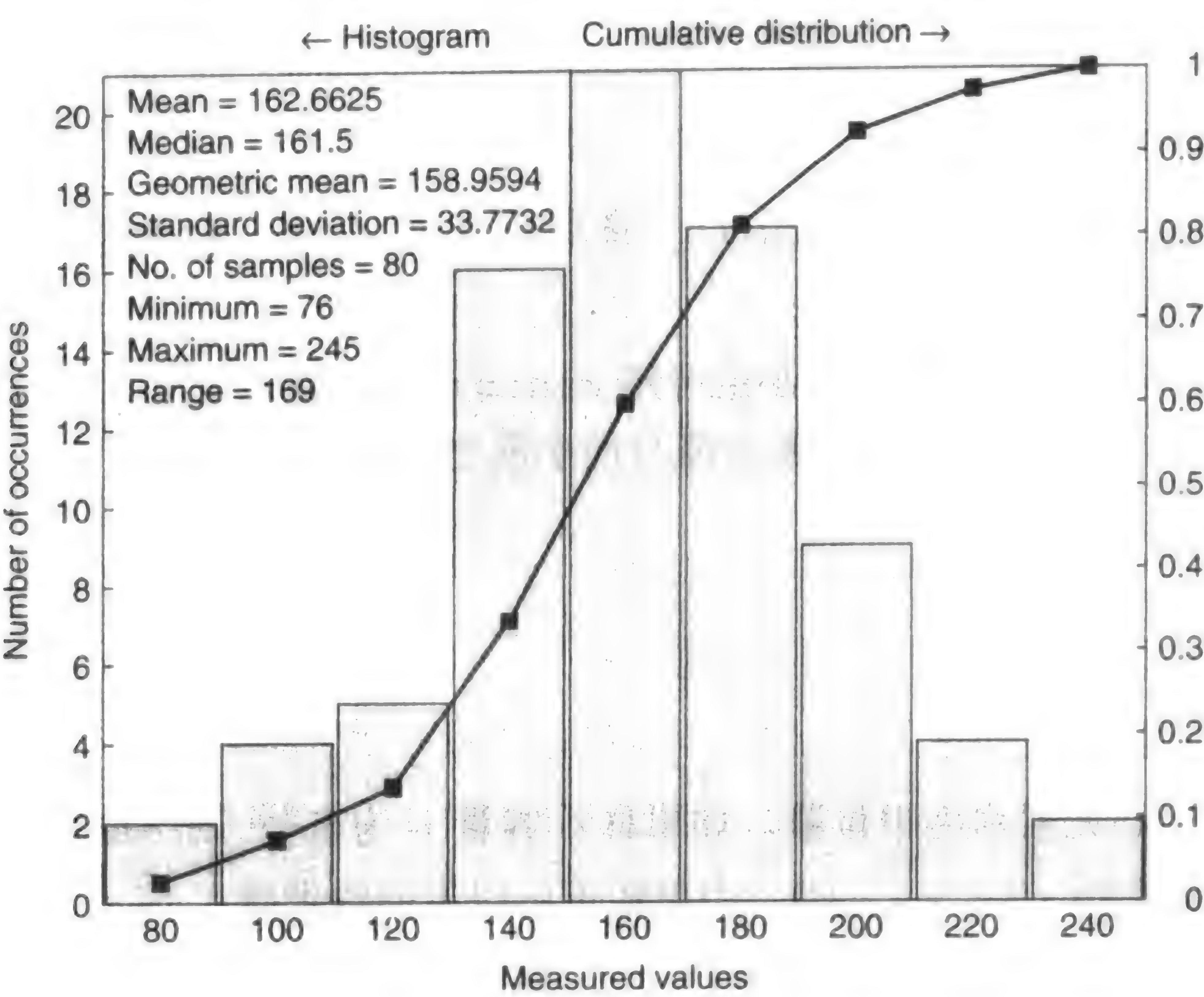


图 14.1 **Dataset141** 的数据直方图及累积分布函数曲线及统计量描述

表 14.1 由 **DataSet141** 构成的数据

105	97	245	163	207	134	218	199
160	196	221	154	228	131	180	178
157	151	175	201	183	153	174	154
190	76	101	142	149	200	186	174
199	115	193	167	171	163	87	176
121	120	181	160	194	184	165	145
160	150	181	168	158	208	133	135
172	171	237	170	180	167	176	158
156	229	158	148	150	118	143	141
110	133	123	146	169	158	135	149

程序为:

```
data = DataSet141;
n = length(data);
b = 80:20:240;
nn = hist(data,b);
maxn = max(nn);
cs = cumsum(nn * maxn/n);
bar(b,nn,0.95,'y')
axis([70,250,0,maxn])
box off
hold on
plot(b,c,'k-s')
title('\leftarrow Histogram     Cumulative distribution \rightarrow')
ylabel('Number of occurrences')
xlabel('Measured values')
text(72,0.97 * maxn,['Mean = 'num2str(mean(data))])
```

```
text(72,0.92 * maxn,['Median = 'num2str(median(data))])
text(72,0.87 * maxn,['Geometric maen = 'num2str(geomean(data))])
text(72,0.82 * maxn,['Standard deviation = 'num2str(std(data))])
text(72,0.77 * maxn,['No. of samples = 'num2str(ldat)])
text(72,.67 * maxn,['Maximum = 'num2str(max(data))])
text(72,.72 * maxn,['Minimum = 'num2str(min(data))])
text(72,.62 * maxn,['Range = 'num2str(range(data))])
plot([70 250],[maxn maxn],'k',[250 250],[0 maxn],'k')
j = 0:.1:1;
lenj = length(j);
text(repmat(251,lenj,1),maxn * j',num2str(j',2))
plot([repmat(248.5,1,lenj);repmat(250,1,lenj)],[maxn * j;maxn * j],'k')
```

执行结果如图 14.1。

尽管区间的中心值可以由 hist 函数计算得出，但程序中仍然定义了这些值，以便更易于控制数据的输出形式。因为 box 函数分别在顶部和右部重复画出横轴和纵轴的刻度线，所以本程序中关闭了 box 函数。而且这里所画的图形为两种不同类型的图，不能用 plotyy 函数实现，而必须使坐标标号独立于刻度线，从而分别画出顶部和右部图形边界线。由此得到的纵轴刻度最大值为 $21j$，即 y 轴的最大值为 21，可以用 axis 函数求出。

如果未定义区间的中心值，则计算出的区间中心值将有所差别，由此画出的直方图也略有不同。该差别也改变了 n_k 值，因此，以下代码的执行结果如图 14.2 所示：

```
[nn,b] = hist(DataSet141,9);
bar(b,nn,0.95,'y');
axis([70,250,0,max(nn)])
```

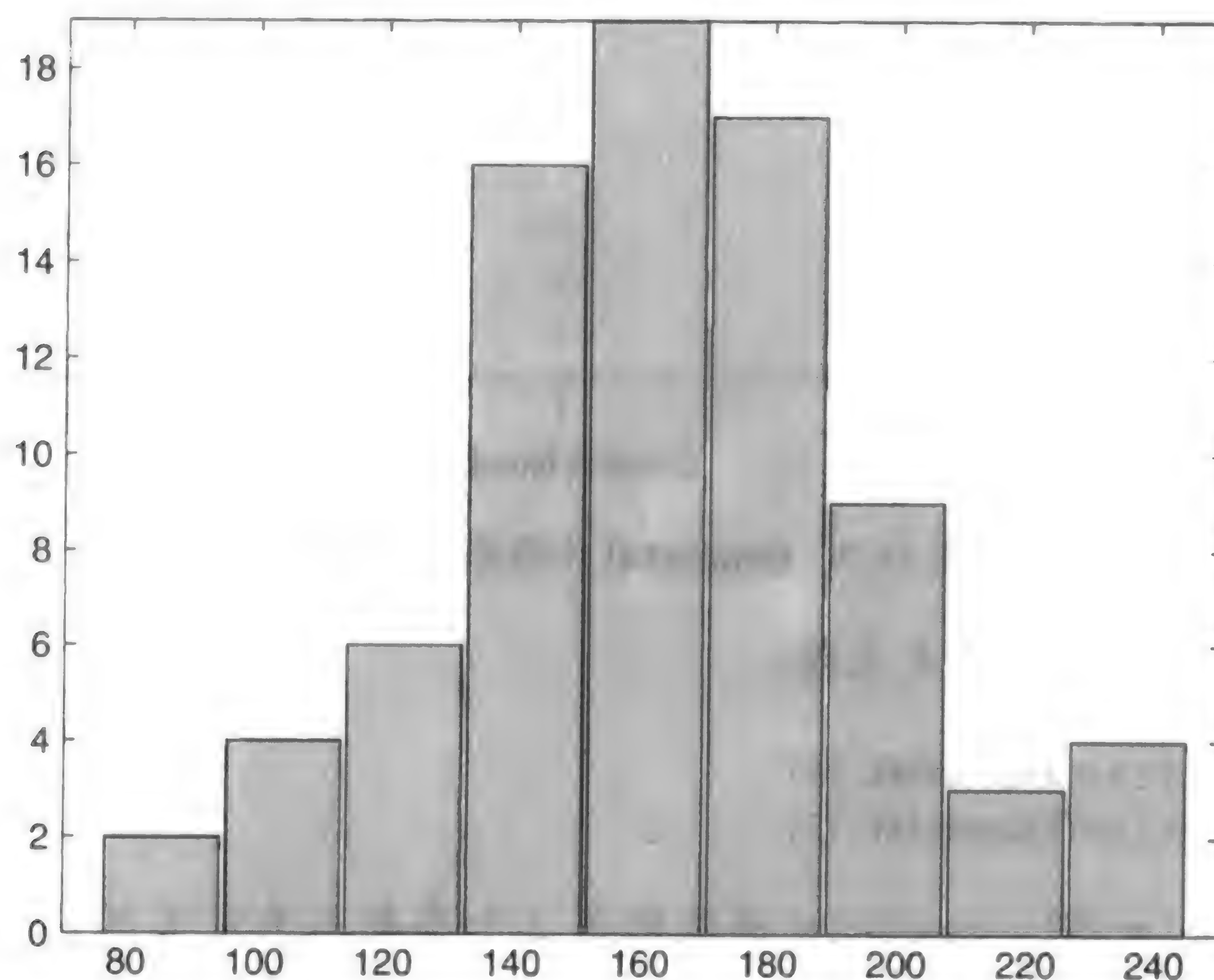


图 14.2　由 hist 函数计算区间中心值并分析 **DataSet141** 数据得到的直方图

图 14.1 和图 14.2 的差别是由区间中心的不同而引起的。第一种情况下区间的中心定义为：

```
b = [80,100,120,140,160,180,200,220,240]
```

而第二种情况由 hist 函数计算出的区间中心值为：

b = [85.38, 104.16, 122.94, 141.72, 160.50, 179.27, 198.05, 216.83, 235.61]

可以看出，在若干个区间内 x_j 的数量有所不同。

另一种方法是用盒状图表示数据。函数 **DataSet141** 的盒状图如图 14.3 所示，该图由以下语句画出：

```
boxplot(DataSet141,'Notch','on')
```

函数的第二个参数设为 1，将产生一沙漏型盒状图，其凹口处为中位数。盒状图的顶部和底部间的区域表示了数据的 50%，其底部为第 1 分位数 q_1，顶部为第 3 分位数 q_3。注意一般盒状图并不是关于中位数对称的。分别从盒状图底部和顶部延伸出去的虚线代表极限值，其范围分别为 $q_1 - 1.5(q_3 - q_1)$和 $q_3 + 1.5(q_3 - q_1)$。位于虚线外的任何点称为界外值，图中用加号表示。盒状图最常用来比较几组数据，如图 14.14(b)中所示。

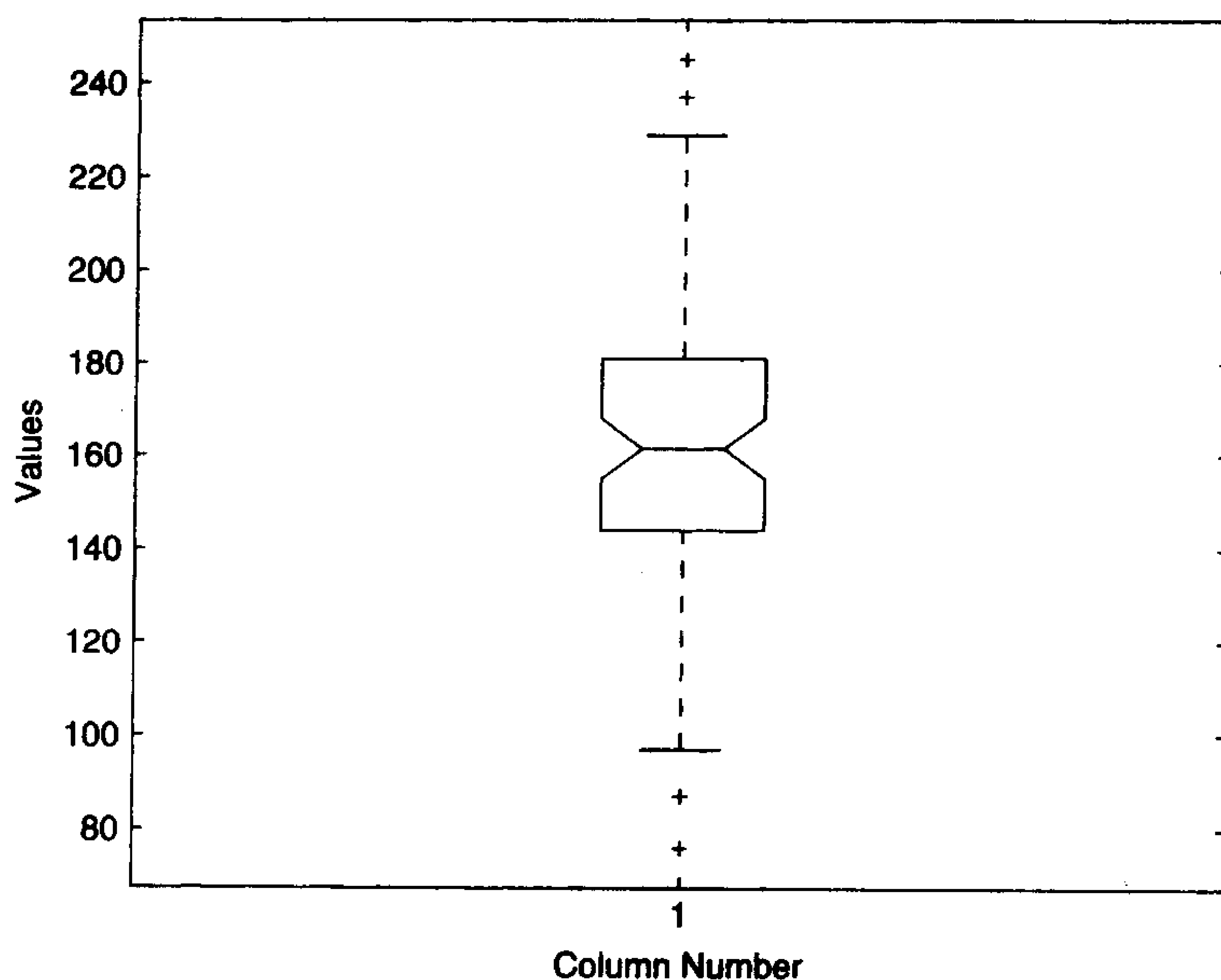


图 14.3 **DataSet141** 中数据的盒状图

q_1 和 q_3 的值分别由以下两式求得：

```
q1 = prctile(DataSet141,25)
q3 = prctile(DataSet141,75)
```

得出 $q_1 = 144$, $q_3 = 181$。prctile 函数的第 2 个参数用来定义所占比例。当比例等于 25%时，即为第 1 分位数。

用以下语句可以确定数据分布平均值是否对称：

```
s = skewness (DataSet141)
```

返回值 $s = -0.0246$，负号意味着数据分布略倾向于左侧。

14.2 概率分布

14.2.1 离散分布

离散随机变量 $X=x$,其中 x 为变量 X 的取值,概率 $P(X)$ 定义为:

$$f(x) = P(X = x) \tag{14.3}$$

对所有 $x, f(x) \geqslant 0$,且:

$$\sum_{\text{all } x_i} f(x_i) = 1 \tag{14.4}$$

$f(x)$称为离散随机变量的概率质量函数。

如果只关注 $X \leqslant x$ 的概率 $P(X \leqslant x)$,则:

$$P(X \leqslant x) = \sum_{x_k \leqslant x} f(x_k) = 1 - \sum_{x_k > x} f(x_k) \tag{14.5}$$

称为累积分布函数。相反,如果只关注 $X \geqslant x$ 的概率 $P(X \geqslant x)$,则:

$$P(X \geqslant x) = \sum_{x_k \geqslant x} f(x_k) \tag{14.6}$$

二项分布

做 n 次重复试验,要求(1)每次试验是独立的,(2)每次试验只有两种可能:“成功”或“失败”,(3)每次试验“成功”的概率 p 保持不变。该概率质量函数服从二项分布:

$$f_b(x) = P(X = x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \quad x = 0,1,\cdots,n \tag{14.7}$$

式中,x 为成功的试验次数。

二项分布的平均值为:

$$\bar{x} = np \tag{14.8a}$$

其标准偏差为:

$$s = \sqrt{np(1-p)} \tag{14.8b}$$

计算二项分布概率质量函数的 MATLAB 函数为:

```
binopdf(x,n,p)
```

计算平均值和方差(s^2)的函数为:

```
[Bmean, Bvariance] = binostat(n,p)
```

其中,$x = 0,1,2,\cdots,n$。

在掷骰子的实验中,任一面成为顶面的概率为 $p = 1/6$。考虑顶面点数为 3 的情况,投掷一次出现 3 点面的概率为:

$$P(X = \text{顶面为 3 点}) = \frac{1!}{1!(0!)} (1/6)^1 (1-1/6)^{1-1} = \frac{1}{6}$$

可由以下语句实现:

```
Pb = binopdf(1,1,1/6)
```

而在两次投掷子中仅有一次顶面为 3 点面的概率为:

```
Pb = binopdf(1,2,1/6)
```

结果为 $Pb = 0.2778 < 1/3$。

在投掷硬币的实验中，正反面的概率均为 $p = 0.5$。投掷 10 次($n = 10$)，出现 4 次为正面($x = 4$)的概率为：

```
Pb = binopdf(4,10,0.5)
```

结果为 $Pb = 0.2051$。

例 14.2 飞机成功起飞的概率问题①

由 16 架飞机组成的空军飞行中队要求做好立即起飞的准备。其中一架飞机不能立即起飞的概率为 20%，重新起飞需要几分钟的时间，因此一架飞机立刻起飞的概率为 0.80。12 架飞机能够成功起飞的概率为：

```
Pb = binopdf(12,16,0.80)
```

计算结果为 $Pb = 0.2001$。

另一方面，至少有 14 架飞机立刻成功起飞的概率[参考式(14.5)和式(14.6)]为：

```
Pb = 1 - binocdf(13,16,0.80)
```

或

```
Pb = sum(binopdf(14:16,16,0.80))
```

计算结果为 $Pb = 0.3518$。

下面为程序，其运行结果如图 14.4 所示。

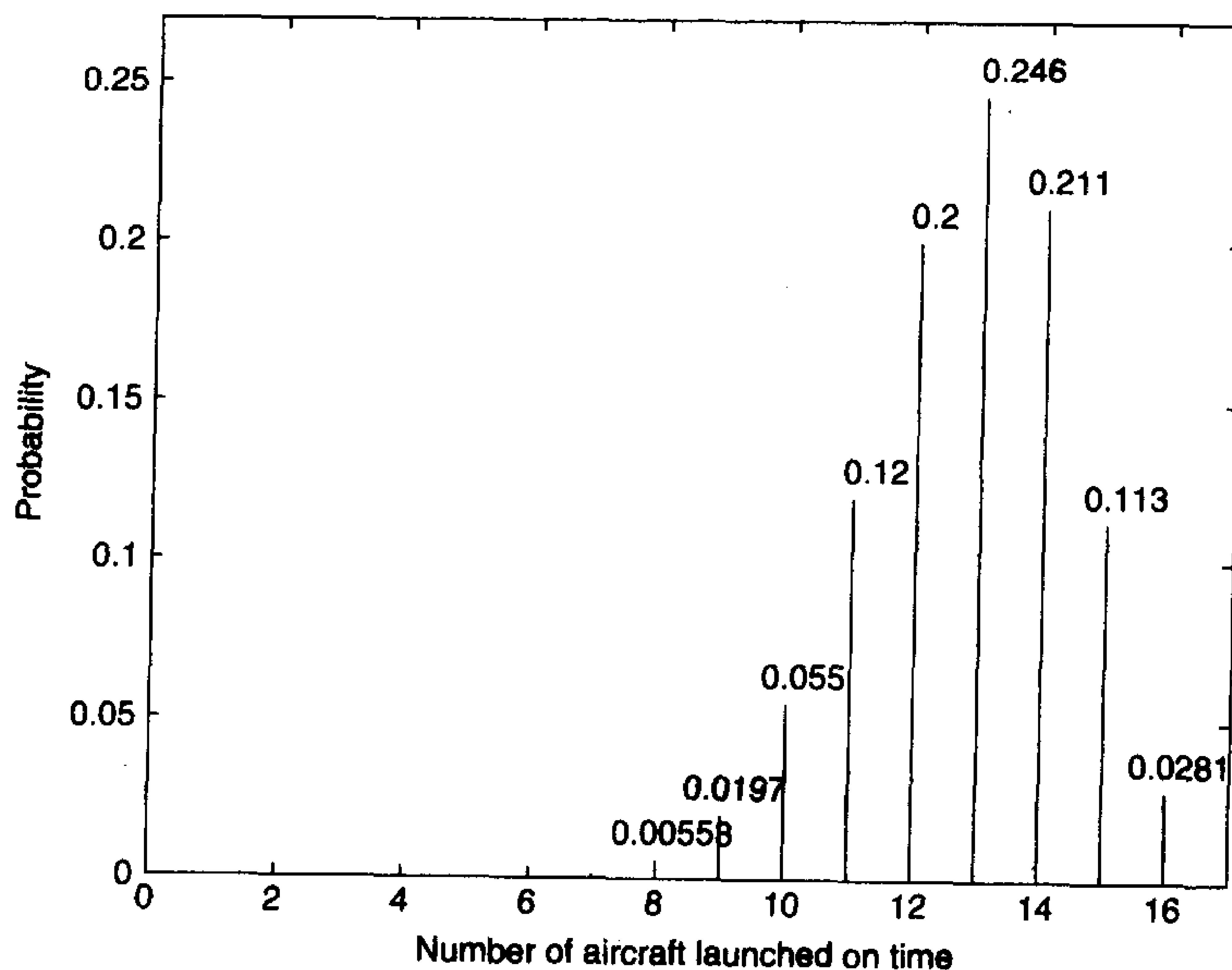


图 14.4　0～16 架飞机按时起飞的概率质量函数

① 参见 A. J. Hayter, *Probability and Statistics for Engineers and Scientists*, PWS Publishing Co., Boston, MA, 1996 年，第 167 页。

```
n=1:16;
Pb=binopdf(n,16,0.80);
plot([n;n],[zeros(1,16);Pb],'k')
text(8-.7:16-.7,Pb(8:16)+.005,num2str(Pb(8:16)',3))
axis([0,17,0,0.27])
xlabel('Number of aircraft launched on time')
ylabel('Probability')
```

泊松分布

假定某一时间段内事件的发生是随机的,该时间段可分为多个更小的子时间段:(1)每个子时间段内事件发生多于一件的概率为零;(2)所有子时间段内事件发生的概率相同,且与子时间段长度成比例;(3)每个子时间段内发生的事件数独立于其他子时间段发生的事件数。这样的一系列事件发生过程叫做泊松过程。如果该时间段内发生事件总数的平均值$\lambda>0$,则其概率质量分布为:

$$f_p(x)=P(X=x)=\frac{e^{-\lambda}\lambda^x}{x!}\qquad x=0,1,2,\cdots \tag{14.9}$$

上式即为同一时间内发生 x 个事件的泊松分布。

泊松分布的平均值为:

$$\bar{x}=\lambda \tag{14.10a}$$

标准偏差为:

$$s=\sqrt{\lambda} \tag{14.10b}$$

泊松分布的概率质量函数由以下语句获得:

```
poisspdf(x,lambda)
```

计算平均值和方差(s^2)的语句为:

```
[Pmean, Pvariance]=poisstat(lambda)
```

例 14.3　医院资源储备

某医院的急救室平均每周收到 46 例心脏病发作病例(每天为 46/7 例),目前医院每天能处理 9 例。医院的工作人员想知道目前其资源能否充足的概率,也就是 $P(X\leqslant 9)$的值。因此可用:

```
Pp=poisscdf(9,46/7)
```

求得结果为 $Pp=0.8712$,由此看出每天还需要增加 13% 的资源。

14.2.2　连续分布

假定 x_1,x_2 为连续随机变量 X 的上下限,则 X 位于$x_1\leqslant X\leqslant x_2$ 的概率 $P(X)$定义为:

$$P(x_1\leqslant X\leqslant x_2)=\int_{x_1}^{x_2}f(x)\mathrm{d}x \tag{14.11a}$$

对所有 x 值,$f(x)\geqslant 0$,且:

$$\int_{-\infty}^{\infty}f(x)\mathrm{d}x=1 \tag{14.11b}$$

$f(x)$ 称为连续随机变量的概率密度函数(pdf,prodability density function)。

其累积分布函数(cdf,cumulative distribution function) $F(x)$为:

$$F(x) = P(X \leqslant x) = \int_{-\infty}^{x} f(u)\mathrm{d}u = 1 - \int_{x}^{\infty} f(u)\mathrm{d}u \tag{14.12}$$

因此,

$$P(X \geqslant x) = 1 - F(x) \tag{14.13}$$

MATLAB 有一大批用于求概率密度的函数。现用其中的两个:正态分布和威布尔(Weibull)分布,其他函数用法与其相似。

正态分布

正态概率分布函数为:

$$f_n(x) = P(X = x) = \frac{1}{\sigma\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad -\infty < x < \infty \tag{14.14}$$

式中,$-\infty < \mu < \infty$和 $\sigma > 0$ 为独立参数。μ 为正态分布的平均值,σ^2 为其方差(σ 为标准偏差)。如有一组数据 $x_j, j = 1,2,\cdots,n$,则其正态概率密度函数 pdf 由以下语句得到:

```
mu = mean(x);
sigma = std(x);
Pn = normpdf(x0, mu, sigma)
```

其中,$x0$ 为所关注的值(或者向量的多个值),$mu = \mu$, $sigma = \sigma$, Pn 的大小与 $x0$ 的大小相对应。μ 和 σ 的估计值可由以下语句得到:

```
[meanx, stddev] = normfit(x)
```

累积分布函数 $\Phi(x)$为:

$$\begin{aligned}\Phi(x) = P(X \leqslant x) &= \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x}\mathrm{e}^{-(u-\mu)^2/2\sigma^2}\mathrm{d}u \qquad -\infty < x < \infty \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{(x-\mu)/\sigma}\mathrm{e}^{-u^2/2}\mathrm{d}u \qquad -\infty < x < \infty\end{aligned} \tag{14.15}$$

或

$$\Phi(z) = P(Z \leqslant z) = \frac{1}{2\pi}\int_{-\infty}^{z}\mathrm{e}^{-u^2/2}\mathrm{d}u \qquad -\infty < z < \infty \tag{14.16}$$

其中,

$$z = (x - \mu)/\sigma \tag{14.17}$$

称为标准正态随机变量,此时 $\mu_z = 0, \sigma_z = 1$。则:

$$\begin{aligned}P(Z \geqslant z) &= 1 - \Phi(z) \\ P(Z \leqslant -z) &= \Phi(-z) \\ P(-z \leqslant Z \leqslant z) &= \Phi(z) - \Phi(-z) \\ P(z_L \leqslant Z \leqslant z_H) &= \Phi(z_H) - \Phi(z_L)\end{aligned} \tag{14.18}$$

由式(14.18)得到的区域如图 14.5 所示。

正态累积分布函数 cdf 由以下语句求得:

```
mu = mean(x);
sigma = std(x);
Pn = normcdf(x0, mu, sigma)
```

其中,$x0$ 为受关注的值(或向量),$mu = \mu$,$sigma = \sigma$,h 的大小与 $x0$ 的大小相对应。如果 x 转化为 z,则 $mu = 0$,$sigma = 1$,这些均为默认值。因此,当 $x \to z$ 时,参数 mu 和 $sigma$ 可省略。

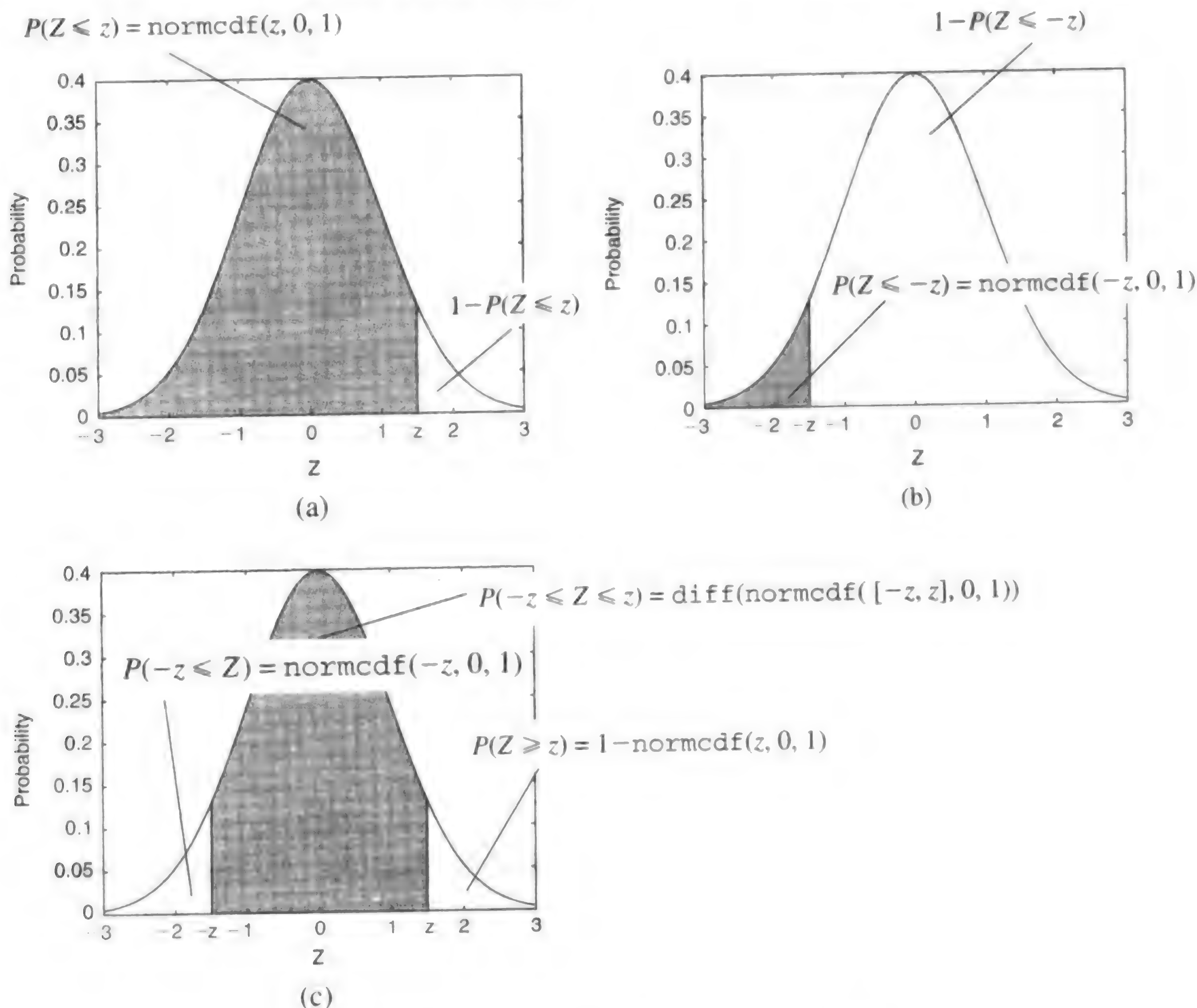


图 14.5　由 normcdf 求出的 cdf 的关系图

(a) $Z \leq z$;(b) $Z \leq -z$;(c) $-z \leq Z \leq z$

参考图 14.5(c)和式(14.18)可以得到 x 在一个取值范围内的概率分布 $P(x_L \leq X \leq x_H)$,即:

```
Pn = diff(normcdf([xL, xH], mean(x), std(x)))
```

例如,在函数 **DataSet141** 定义的数据中,介于 120 和 200 之间的测量数据的概率为:

```
Pn = diff(normcdf([120,200], mean(DataSet141), std(DataSet141)))
```

其结果为 $Pn = 0.7623$。此值与图 14.1 中得到的估计值非常接近。

在一些例子中,有时希望求出 $\Phi(x)$ 的反函数,即:

$$x = \Phi^{-1}[P(X \leq x)] \tag{14.19}$$

可由以下函数实现:

```
norminv(p, mean(x), std(x))
```

其中,p 为累积概率,如图 14.5(a)中的阴影部分所示。

为确定一组数据能否用正态概率密度函数表示,通常需要在正态概率图中画出这些数据。

正态概率图的纵坐标(y 轴)的刻度由累积正态分布函数划分。在正态概率坐标上画图与在对数刻度坐标上画图类似。在对数坐标图上指数函数将显示为一条直线;与之类似,在以正态累积概率函数坐标上画出服从正态分布的过程将显示为一条直线。换句话说,位于平均值 μ 两侧标准偏差 σ 处的累积概率值服从正态分布,其值分别为 $P(X \leqslant \mu+\sigma)=0.84$ 和 $P(X \leqslant \mu-\sigma)=0.16$。平均值处的累积概率值 $P(x \leqslant \mu)=0.5$,转换到概率图上的三组坐标($\mu-\sigma$,0.16),(μ,0.5)和($\mu+\sigma$,0.84)对应的三个点位于一条直线上。μ 和 σ 分别由式(14.1)和式(14.2)估算,并用 MATLAB 函数 mean 和 std 计算。

在概率坐标上画图的过程如下所述。有 m 个数据 y_i, $i=1,2,\cdots,m$,将其按照从小(最负值)到大(最大正数)顺序排序,将最小值编号为 1,依次为 2,3,等等,最大值编号为 m。这些排好序的数据表示为 $w_j, j=1,2,\cdots,m$。对每个指定累积概率 $w_j, j=1,2,\cdots,m$,即概率 $P(w \leqslant w_j)$。概率图所画数据值的坐标为(ω_j,$(j-0.5)/m$)。如果只有线性坐标图,该点坐标变为(w_j, z_j),其中,z_j = norminv($(j-0.5)/m$)。执行上述计算并画出图形的函数为:

```
normplot(y)
```

其中,$y=[y_1\ y_2 \cdots y_m]$。图中直线由 y_i 和 z_j 的第 1 分位数和第 3 分位数组成的坐标确定,可参照 14.1 节中的 q_1 和 q_3 的定义及图 14.3 中的解释。

例 14.4　正态分布数据的验证

再次利用 **DataSet141** 函数所定义的数据重新画出对应的直方图,并在柱条上重叠画出对应的正态概率密度函数曲线。可由以下语句实现:

```
histfit(DataSet141,9)
colormap([1,1,1])
```

结果如图 14.6 所示。colormap 函数将柱条颜色改为白色。然后验证数据是否服从正态分布。用以下语句可得到如图 14.7 所示的结果:

```
normplot(DataSet141)
```

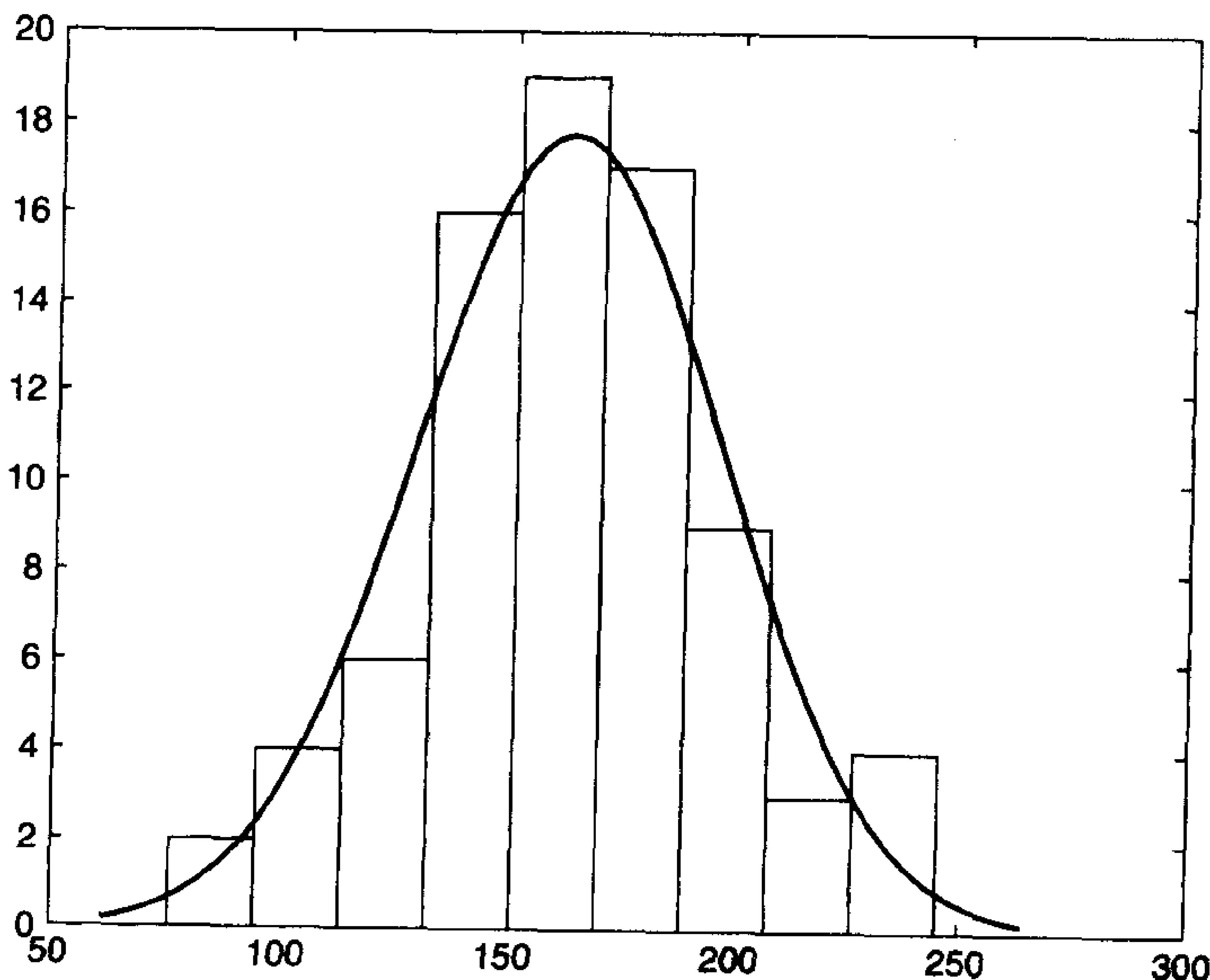


图 14.6　叠加正态概率密度函数(pdf)曲线直方图

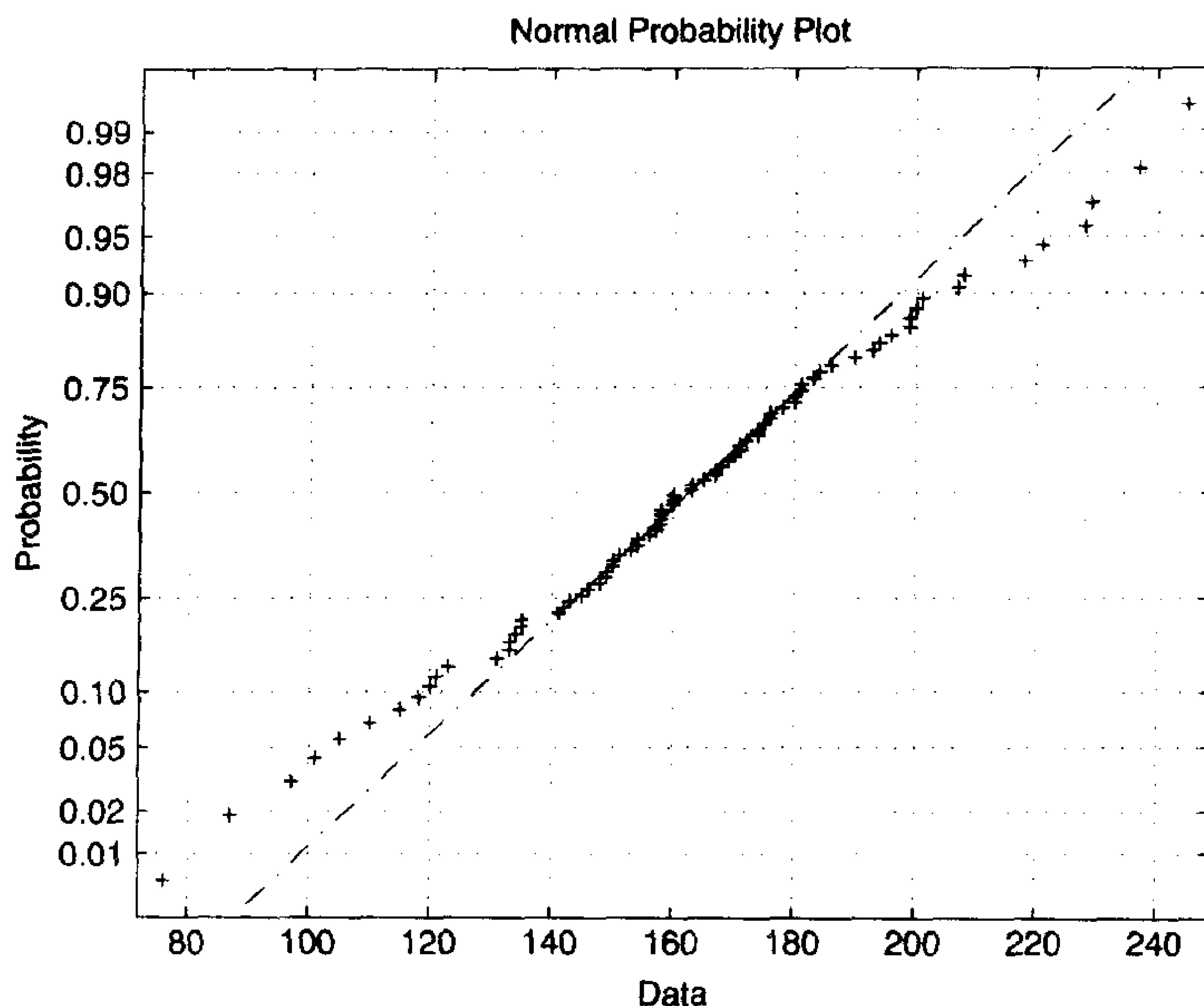


图 14.7 **DataSet141**所定义数据的正态累积概率图

从图中可看出有相当多的数据位于直线附近,可以得出结论认为该数据近似服从正态分布。如果承认这些数据服从正态分布,则90%的数据分布值可确定出来。参照图14.5(a)和图14.5(c)以及式(14.19),用 norminv 函数求解如下:

```
zh = norminv(.95, mean(DataSet141),std(DataSet141))
zl = norminv(.05, mean(DataSet141),std(DataSet141))
```

得到结果为 $z_h = 218.2145$ 及 $z_l = 107.1105$。为验证此结果,现求出这两个界限概率之差,如下所述:

```
m = mean(DataSet141);
s = std(DataSet141);
zh = norminv(.95,m,s);
zl = norminv(.05,m,s);
ph = normcdf(zh,m,s);
pl = normcdf(zl,m,s);
```

执行结果为:$p_h - p_l = 0.9500 - 0.0500 = 0.90$。

例 14.5 正态分布与泊松分布和二项分布接近的情况

当 $np>5$ 且 $n(1-p)>5$ 时,正态分布非常近似于二项分布。当 $\lambda>5$ 时,$P(X\leqslant x)$的分布也非常近似于泊松分布。对于二项分布,标准正态随机变量为[参考式(14.8)]:

$$z_b = \frac{X - np}{\sqrt{np(1-p)}}$$

对于泊松分布[参考式(14.10)]为:

$$z_p = \frac{X - \lambda}{\sqrt{\lambda}}$$

为说明泊松分布与正态分布的近似性,再参考例14.3。例14.3中得出某医院每天治疗9

例以下心脏病患者的概率为 0.8712。现在用正态分布重新计算,已知 $\mu = 46/7$ 及 $\sigma = \sqrt{46/7}$,则:

```
P = normcdf((9 - 46/7)/sqrt(46/7),0,1)
```

结果为 $P = 0.8283$,在精确值的 4.9% 范围内。为直观地显示出这种近似关系,运行下面的程序,可以绘出例 14.2 中的泊松分布图形以及由上述表达式得到的近似图形:

```
x = 1:16;
y = linspace(00,16,50);
yPoisson = poisspdf(x,46/7);
NormApprox = normpdf(y,46/7,sqrt(46/7));
plot([x;x],[zeros(1,16);yPoisson],'k',y,NormApprox,'k')
xlabel('Number of aircraft launched on time')
ylabel('Probability density')
```

其执行结果如图 14.8 所示。

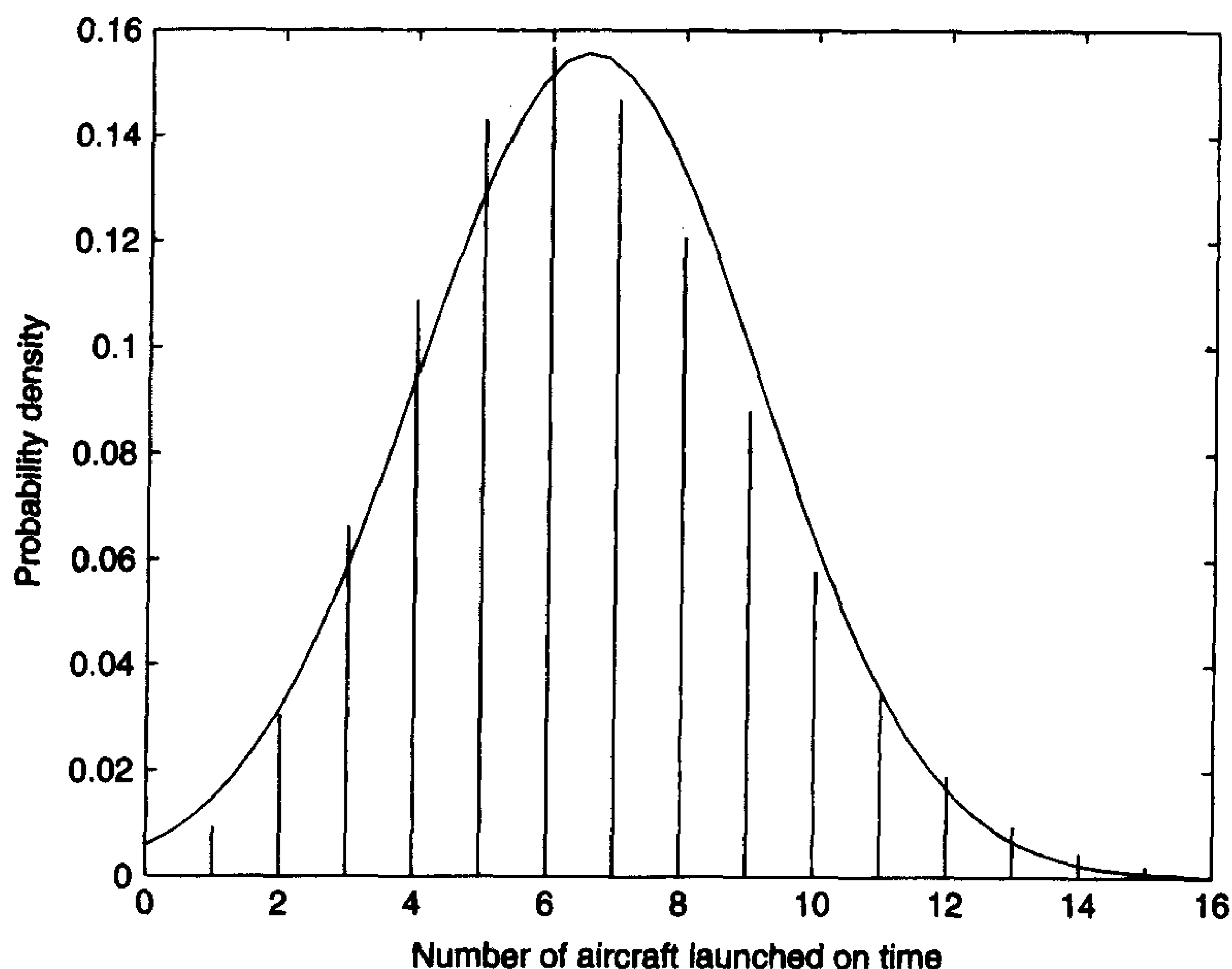


图 14.8 例 14.3 中的泊松分布及其近似的正态分布图

威布尔分布

威布尔概率分布函数为:

$$f_w(x) = \alpha\beta x^{\beta-1} e^{-\alpha x^{\beta}} \quad x > 0 \tag{14.20}$$

式中,$\alpha > 0$ 为尺度参数,$\beta > 0$ 为形状参数(另一种常用表示形式为经 $\alpha = \delta^{-\beta}$ 变换后得到的形式)。当 $\beta = 1$ 时,式(14.20)变为指数分布;当 $\beta = 2$ 时,变为瑞利(Rayleigh)分布。该概率密度函数的平均值为:

$$\mu_w = \alpha^{-1/\beta}\Gamma\left(1 + \frac{1}{\beta}\right)$$

方差为:

$$\sigma_w^2 = \alpha^{-2/\beta}\Gamma\left(1+\frac{2}{\beta}\right) - \alpha^{-2/\beta}\left[\Gamma\left(1+\frac{1}{\beta}\right)\right]^2$$

式中，$\Gamma(x)$为伽马函数。威布尔概率密度函数由以下语句实现：

```
Pw = wblpdf(x, alpha, beta)
```

其中，$alpha = \alpha$，$beta = \beta$，*Pw* 的大小与 x 的大小相对应。其平均值和方差由以下语句得到：

```
[muW, VarW] = wblstat(alpha, beta)
```

累积分布函数 $F_w(x)$为：

$$F_w(x) = P(X \leqslant x) = 1 - e^{-\alpha x^\beta} \quad x > 0 \tag{14.21}$$

则由以下语句得到：

```
Wcdf = wblcdf(x, alpha, beta)
```

其中，*Wcdf* 的大小与 x 的大小相对应。

有时需求出 $F(x)$的反函数，即：

$$x = F_w^{-1}[P(X \leqslant x)] \tag{14.22}$$

可用以下函数实现：

```
wblinv(p, alpha, beta)
```

其中，p 为累积概率分布值。

例 14.6　对服从威布尔分布的数据进行验证

表 14.2 所列为元件寿命的分类数据，下面创建函数 **DataSet142** 来存放这些数据：

```
function d = DataSet142
d = [72 82 97 103 113 117 126 127 127 139 154 159 199 207]';
```

在图上画出这些数据以确定是否能用威布尔分布描述。用 wblplot 函数实现，其结果如图 14.9 所示：

```
wblplot (DataSet142)
```

由图中可看出，这些数据用威布尔分布描述非常合适。因此，将采用威布尔分布模型来表示这些数据。

表 14.2　元件寿命分类数据——DataSet142

元件寿命	元件寿命
72	127
82	127
97	139
103	154
113	159
117	199
126	207

使用 ab = wblfit(x)确定 α 值和 β 值。

其中,$ab(1)=\alpha$,$ab(2)=\beta$。求 α,β 的大小及平均值和标准方差的程序为:

```
ab = wblfit(DataSet142)
[muW,varW] = wblstat(ab(1),ab(2))
sigW = sprt(varW)
```

执行程序得 $\alpha=144.27$,$\beta=3.644$,$\mu_W=130.09$,$\sigma_W=39.70$。

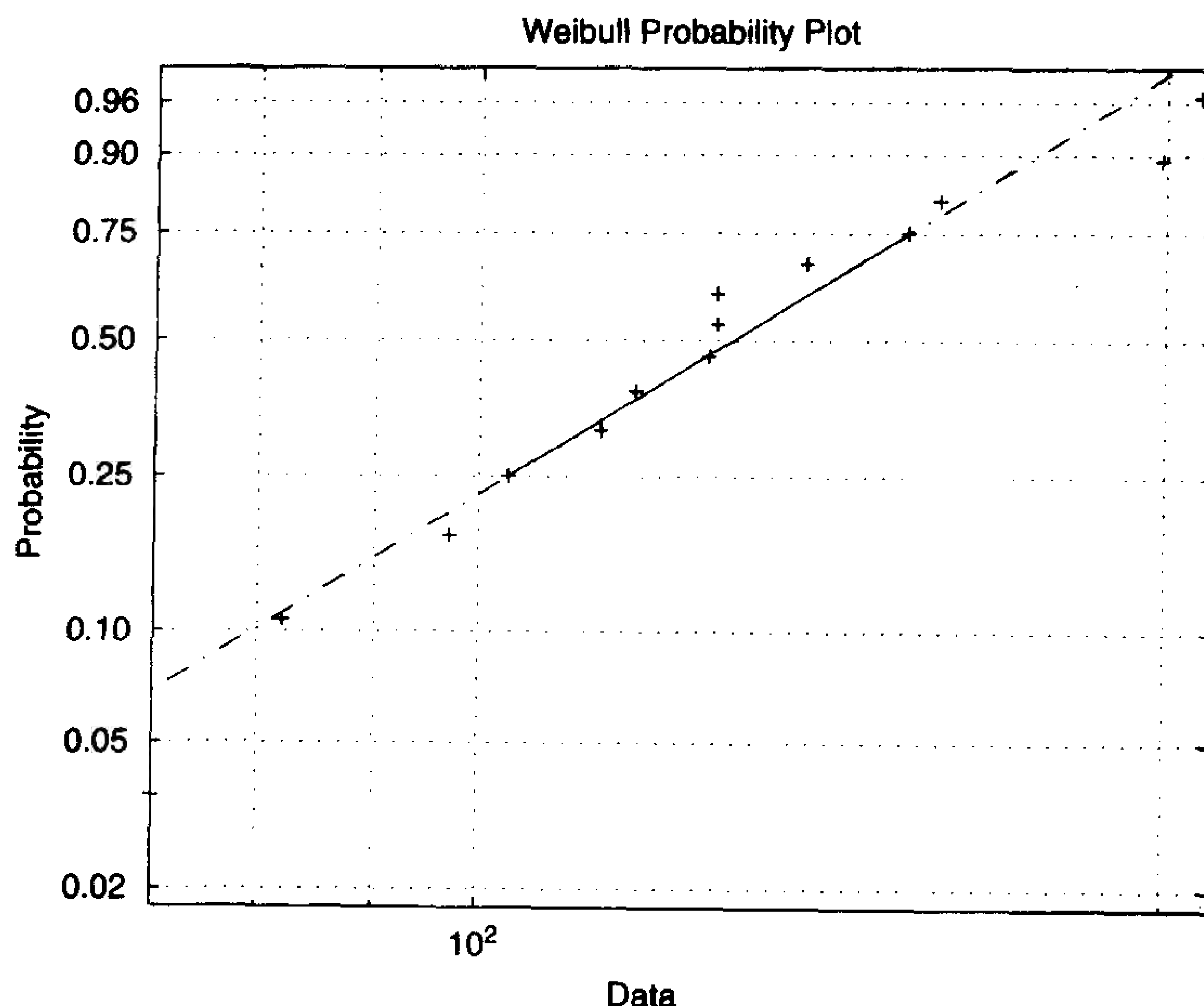


图 14.9　DataSet142 数据的威布尔累积概率图

假定为一正态分布,则求平均值和标准方差由以下语句实现:

```
muN = mean(DataSet142)
sigmaN = std(DataSet142)
```

运行结果为 $\mu_{norm}=130.14$ 和 $\sigma_{norm}=39.39$。

运行下面的代码将画出表 14.2 中数据的威布尔概率密度函数和正态密度函数的关系图,以便于两者进行比较。程序如下:

```
ab = wblfit(DataSet142);
xx = linspace(50,200,50);
yW = wblpdf(xx,ab(1),ab(2));
yN = normpdf(xx,mean(DataSet142),std(DataSet142));
plot(xx,yW,'k-',xx,yN,'k--')
legend('Weibull','Normal')
xlabel('x')
ylabel('Probability density function')
```

执行结果如图 14.10 所示。

现在要求确定元件寿命少于 100 小时的概率,程序如下:

```
ab = wblfit(DataSet142);
```

```
p = wblcdf(100,ab(1),ab(2))
```

其结果为 $p = 0.2312$，或 23.1%。

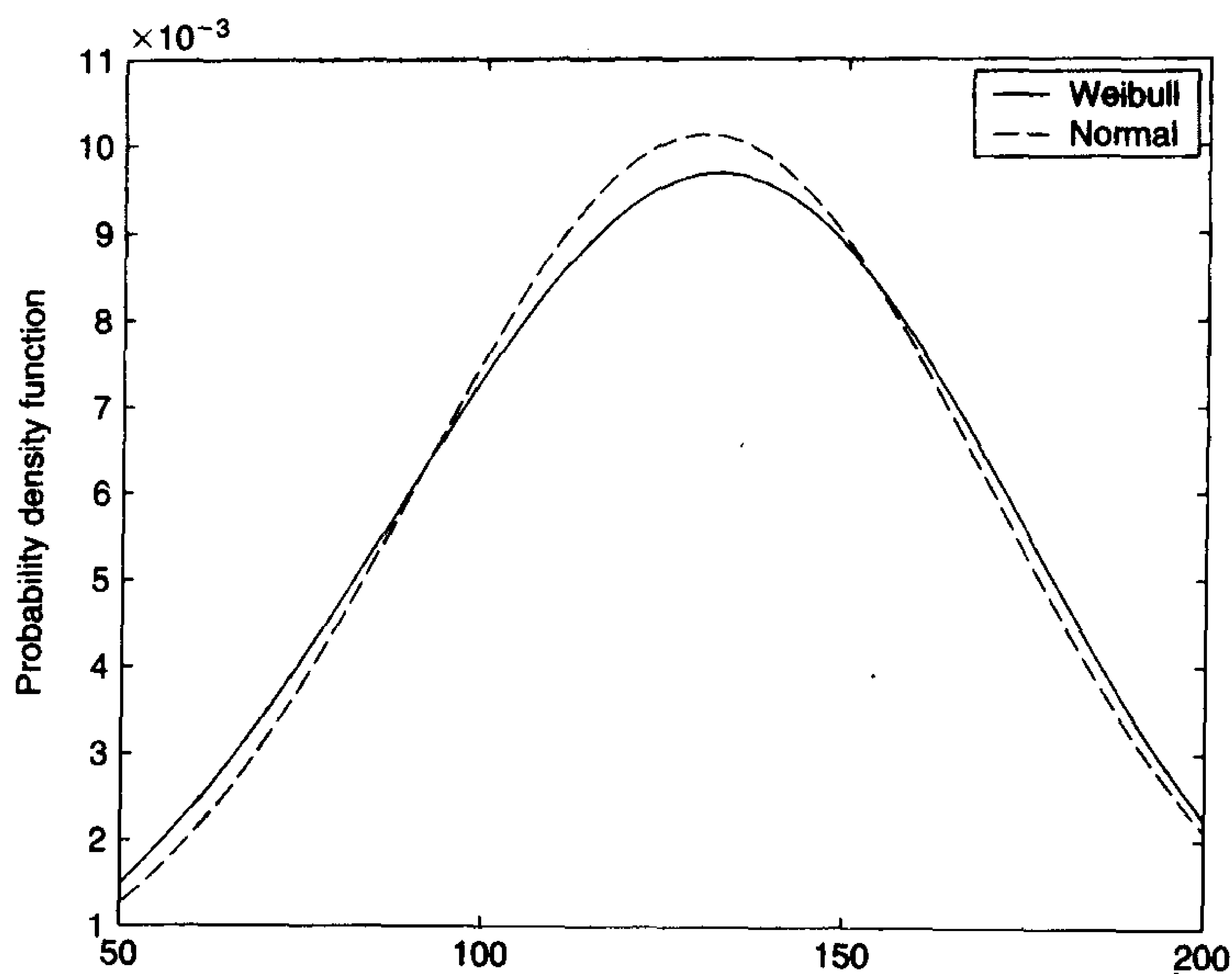


图 14.10　由 DataSet142 数据得出的威布尔函数和正态概率密度函数的比较

14.3　置信区间

令 θ 为 n 个数据样本中的某一统计量(例如平均值、方差、均差等)。人们所关心的是下式中 l 和 u 的值：

$$P(l \leqslant \theta \leqslant u) = 1 - \alpha$$

式中，$0 < \alpha < 1$。上式说明，在容量为 n 的样本中找出包含 θ 真值的区间的概率为 $1 - \alpha$。该区间：

$$l \leqslant \theta \leqslant u$$

称做置信度为 $100(1 - \alpha)\%$ 的 θ 的双限置信区间。l 和 u 分别称做置信上限和置信下限。与之类似，置信度为 $100(1 - \alpha)\%$ 的下限置信区间为：

$$l \leqslant \theta$$

置信度为 $100(1 - \alpha)\%$ 的上限置信区间为：

$$\theta \leqslant u$$

置信上限和下限值取决于样本的分布，还与总体标准偏差是否为已知有关。表 14.3 中总结了几种常用的确定置信上下限的关系式，其中用到以下定义：

μ 和 σ 分别为真实平均值和标准偏差值

$\bar{x}$ 和 s^2 分别由式(14.1)和式(14.2)确定

$t_{\alpha,n-1}$ 为由函数 tinv 求得的服从 $n-1$ 自由度的 t 分布值

$z_{\alpha/2}$ 为由函数 norminv 求得的正态分布值

$\chi^2_{\alpha/2,n-1}$为由函数 chi2inv 得到的 $n-1$ 自由度的 χ^2 分布值

$f_{\alpha/2,n-1,m-1}$为由函数 finv 得到的 $n-1$ 自由度和 $m-1$ 自由度的 f 分布值

表 14.3 几种置信区间过程的总结

问题类型	统计量 $\hat{\theta}$	100(1-α)% 置信区间 $\hat{\theta}-q\leq\theta\leq\hat{\theta}+q$：$\theta$	q		情况
已知 σ^2 的平均值问题	$\bar{x}$	μ	$z_{\alpha/2}\sigma/\sqrt{n}$		1
已知 σ_1^2 和 σ_2^2 的均差问题	$\bar{x}_1-\bar{x}_2$	$\mu_1-\mu_2$	$z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}$		2
未知 σ^2 的平均值问题	$\bar{x}$	μ	$t_{\alpha/2,n-1}s/\sqrt{n}$		3
未知 $\sigma_1^2=\sigma_2^2$ 的均差问题	$\bar{x}_1-\bar{x}_2$	$\mu_1-\mu_2$	$t_{\alpha/2,n_1+n_2-2}s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$		4
未知 $\sigma_1^2\neq\sigma_2^2$ 的均差问题	$\bar{x}_1-\bar{x}_2$	$\mu_1-\mu_2$	$t_{\alpha/2,v}\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}$		5
	$\hat{\theta}$	100(1-α)% 置信区间 $q_1\hat{\theta}\leq\theta\leq q_2\hat{\theta}$：$\theta$	q_1	q_2	
方差	S^2	σ^2	$\frac{n-1}{\chi^2_{\alpha/2,n-1}}$	$\frac{n-1}{\chi^2_{1-\alpha/2,n-1}}$	6
方差比值	$\frac{s_1^2}{s_2^2}$	$\frac{\sigma_1^2}{\sigma_2^2}$	$\frac{1}{f_{\alpha/2,n_1-1,n_2-1}}$	$f_{\alpha/2,n_2-1,n_1-1}$	7

第 4 种情况中的 s_p 为：

$$s_p=\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

第 5 种情况中的 v 为：

$$v=\left(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}\right)^2\left[\frac{(s_1^2/n_1)^2}{n_1+1}+\frac{(s_2^2/n_2)^2}{n_2+1}\right]^{-1}-2$$

情况 3 和情况 7 的置信区间

对第 3 种情况，双限置信区间为：

$$\bar{x}-t_{\alpha/2,n-1}s/\sqrt{n}\leqslant\mu\leqslant\bar{x}+t_{\alpha/2,n-1}s/\sqrt{n}$$

对于第 7 种情况，双限置信区间为：

$$\frac{s_1^2}{s_2^2}\frac{1}{f_{\alpha/2,n_1-1,n_2-1}}\leqslant\frac{\sigma_1^2}{\sigma_2^2}\leqslant\frac{s_1^2}{s_2^2}f_{\alpha/2,n_2-1,n_1-1}$$

注意，上式中 f 的自由度顺序是相反的。

下面说明表 14.3 中情况 3 和情况 7 的双限置信区间的求解。

例 14.7 双限置信区间

情况 3：

对于第 3 种情况，考虑表 14.1 中的数据，该数据放于函数 **DataSet141** 内。如令置信度为 95%，求置信区间的程序如下所述：

```
meen = mean(DataSet141);
L = length(DataSet141);
q = std(DataSet141) * tinv(0.975,L-1)/sqrt(L);
disp(['Sample mean = 'num2str(meen)])
disp('Confidence interval for sample mean at 95% confidence level-')
disp([' 'num2str(meen-q)' <= Sample mean <= 'num2str(meen + q)])
```

执行结果在 MATLAB 命令窗口中显示下列数据：

```
Sample mean = 162.6625
Confidence interval for sample mean at 95% confidence level-
  155.1466 <= Sample mean <= 170.1784
```

另一个求置信区间的函数为 ttest，将在 14.4 节介绍。

情况 7:

对于第 7 种情况，考虑表 14.4 中的两列数据，这些数据由函数 **DataFci** 定义如下：

```
function [set1,set2] = DataFci
set1 = [41.60 41.48 42.34 41.95 41.86 42.18 41.72 42.26 41.81 42.04];
set2 = [39.72 42.59 41.88 42.00 40.22 41.07 41.90 44.29];
```

表 14.4 第 7 种情况中用到的数据

Set 1	Set 2
41.60	39.72
41.48	42.59
42.34	41.88
41.95	42.00
41.86	40.22
42.18	41.07
41.72	41.90
42.26	44.29
41.81	
42.04	

程序如下：

```
[data1,data2] = DataFci;
r = var(data1)/var(data2);
L1 = length(data1);
L2 = length(data2);
q2 = r * finv(0.975,L2-1,L1-1);
q1 = r/finv(0.975,L1-1,L2-1);
disp(['Ratio of sample variances = 'num2str(r)])
disp('Confidence interval for ratio of sample variances at 95% confidence level-')
disp([' 'num2str(q1)' <= Ratio of sample variances <= 'num2str(q2)])
```

执行后在 MATLAB 命令窗口中显示以下数据：

```
Ratio of sample variances = 0.039874
Confidence interval for ratio of sample variances at 95% confidence level -
  0.0082672 <= Ratio of sample variances <= 0.16736
```

14.4 假设检验

在工程上,许多情况下需要接受或拒绝某些参数的描述(假设)。统计假设可以认为是对一个或多个总体参数的描述,总体是指所重点观察的全部,而单个样本是总体的子集。如果用概率分布表示总体,则统计假设可以认为是对总体的统计分布的描述。

假定参数 θ 是从某个总体的 n 个样本中得到的统计量,现在要判断 θ 是否等于 θ_0。假设检验过程可分为以下步骤:

1. 提出原假设 H_0
2. 建立合适的检验统计量 q_0
3. 选择置信度[θ 的置信度为 $100(1-\alpha)\%$]
4. 将检验统计量 q_0 与期望自然发生的统计量 q 的值相比较

根据 q_0 和 q 的不同数量级可以确定接受或是放弃原假设。如果放弃原假设,则接受了另一假设,用 H_1 表示。

有下列三种情况需要考虑:

$H_0:\theta=\theta_0$　　$H_0:\theta=\theta_0$　　$H_0:\theta=\theta_0$

$H_1:\theta\neq\theta_0$　　$H_1:\theta>\theta_0$　　$H_1:\theta<\theta_0$

每种情况对应统计量 $q_0(n,\alpha)$和 $q(n,\alpha)$。表 14.5 总结了几种对假设的检验过程,与表 14.3 中所列的置信区间过程相对应。表 14.5 中各个量的定义可参考 14.3 节。

在假设检验过程中可能会得出两种类型的错误结论:

第 1 种错误:原假设 H_0 为真时被放弃

第 2 种错误:原假设 H_0 为假时被接受——此时 $\theta=\theta_1$

产生第 1 种错误的概率用 α 表示,产生第 2 种错误的概率用 β 表示。

现在普遍用一种称做 p 值的量代替置信参数 α。p 值越小,原假设成立的可能性越小。现在用表 14.5 中的情况 2、情况 4 和情况 7 来解释这些概念。

表 14.5 几种假设检验过程

原假设 H_0	另一种假设 H_1	放弃 H_0 的条件	检验设计	MATLAB 函数	情况
$\mu=\mu_0$ (已知 σ)	$\mu\neq\mu_0$ $\mu>\mu_0$ $\mu<\mu_0$	$\lvert z_0\rvert>z_{\alpha/2}$ $z_0>z_\alpha$ $z_0<-z_\alpha$	$z_0=\dfrac{\bar{x}-\mu_0}{\sigma/\sqrt{n}}$	ztest	1
$\mu=\mu_0$ (未知 σ)	$\mu\neq\mu_0$ $\mu>\mu_0$ $\mu<\mu_0$	$\lvert t_0\rvert>t_{\alpha/2,n-1}$ $t_0>t_{\alpha,n-1}$ $t_0<-t_{\alpha,n-1}$	$t_0=\dfrac{\bar{x}-\mu_0}{s/\sqrt{n}}$	ttest	2
$\mu_1=\mu_2$ (已知 σ_1 和 σ_2)	$\mu_1\neq\mu_2$ $\mu_1>\mu_2$ $\mu_1<\mu_2$	$\lvert z_0\rvert>z_{\alpha/2}$ $z_0>z_\alpha$ $z_0<-z_\alpha$	$z_0=\dfrac{\bar{x}_1-\bar{x}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1}+\dfrac{\sigma_2^2}{n_2}}}$		3

（续表）

原假设 H_0	另一种假设 H_1	放弃 H_0 的条件	检验设计	MATLAB 函数	情况
$\mu_1=\mu_2$ (未知 $\sigma_1=\sigma_2$)	$\mu_1\neq\mu_2$ $\mu_1>\mu_2$ $\mu_1<\mu_2$	$\|t_0\|>t_{\alpha/2,n_1+n_2-2}$ $t_0>t_{\alpha,n_1+n_2-2}$ $t_0<t_{\alpha,n_1+n_2-2}$	$t_0=\dfrac{\bar{x}_1-\bar{x}_2}{s_p\sqrt{\dfrac{1}{n_1}+\dfrac{1}{n_2}}}$	ttest2	4
$\mu_1=\mu_2$ (未知 $\sigma_1\neq\sigma_2$)	$\mu_1\neq\mu_2$ $\mu_1>\mu_2$ $\mu_1<\mu_2$	$\|t_0\|>t_{\alpha/2,v}$ $t_0>t_{\alpha,v}$ $t_0<-t_{\alpha,v}$	$t_0=\dfrac{\bar{x}_1-\bar{x}_2}{\sqrt{\dfrac{s_1^2}{n_1}+\dfrac{s_2^2}{n_2}}}$		5
$\sigma^2=\sigma_0^2$	$\sigma^2\neq\sigma_0^2$ $\sigma^2>\sigma_0^2$ $\sigma^2<\sigma_0^2$	$\chi_0^2>\chi_{\alpha/2,n-1}^2$ $\chi_0^2>\chi_{\alpha,n-1}^2$ $\chi_0^2<\chi_{1-\alpha,n-1}^2$	$\chi_0^2=\dfrac{(n-1)s^2}{\sigma_0^2}$		6
$\sigma_1^2=\sigma_2^2$	$\sigma_1^2\neq\sigma_2^2$ $\sigma_1^2>\sigma_2^2$	$f_0>f_{\alpha/2,n_1-1,n_2-1}$ 或 $f_0<f_{1-\alpha/2,n_1-1,n_2-1}$ $f_0>f_{\alpha,n_1-1,n_2-1}$	$f_0=\dfrac{s_1^2}{s_2^2}$		7

例 14.8　平均值和方差的统计重要性的检验

情况 2:

用表 14.1 中函数 **DataSet141** 包含的数据对情况 2 加以分析。在置信度为 95% 时确定样本平均值与数值 168($\mu_0=168$)之间是否存在显著性差异。这里假设：

$$H_0: \mu=168$$

$$H_1: \mu\neq168$$

用函数 ttest 分析该假设的合理性。函数 ttest 的用法如下：

```
[h,p,ci] = ttest(Data,muzero,alpha)
```

其中，*Data* 为数据，*muzero* $=\mu_0$，*alpha* $=\alpha$，H_0 成立时 $h=0$，H_1 成立时 $h=1$。对双限置信区间 p 值的计算方法为：

```
p=2*(1-tcdf(t0,n-1));
```

其中，$t0=t_0$ 在表 14.5 中的第 2 行第 4 列定义，$ci(1)=l$ 及 $ci(2)=u$ 分别为置信下限和置信上限。则：

```
[h,p,ci] = ttest(DataSet141,168,0.05)
```

其结果为 $h=0$，即不能放弃原假设，$p=0.1614$，$ci(1)=155.1466$，$ci(2)=170.1784$。在 14.3节的情况 3 中，已求出 $\bar{x}=162.6625$，而且当置信度为 95% 时，置信区间为$155.1466\leqslant\bar{x}\leqslant170.1784$。因为假设平均值取 168，该数值位于上述置信区间内，所以应该推断出原假设将不应被放弃。事实上，由 p 值也可以看出只有 $100(1-0.1614)=83.9\%$ 的置信度，比期望值 95% 低。另一方面，如原假设为：

$$H_0: \mu=175$$

$$H_1:\ \mu \neq 175$$

则:

```
[h,p,ci] = ttest(DataSet141,175,0.05)
```

求得 $h=1$,即应放弃原假设而接受 H_1 假设,$p=0.0016$ 及 $ci(1)=155.1466$, $ci(2)=170.1784$。换句话说,可以 99.84% [即(100(1-0.0016)]地相信 **DataSet141** 中数据的平均值不等于 175。

情况 4:

在情况 4 中仍以函数 **DataFci** 定义的表 14.4 中的数据为例,在 95% 的置信度上确定这些样本的平均值是否有显著差异。假设:

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 \neq \mu_2$$

现用 ttest2 分析假设的合理性。函数 ttest2 的用法如下:

```
[h,p,ci] = ttest2(x1,x2,alpha)
```

其中,$x1$ 和 $x2$ 为数据,$alpha=\alpha$,H_0 成立时 $h=0$,H_1 成立时 $h=1$。对双限置信区间,p 值的计算方法如下:

```
p=2*(1-tcdf(t0,n-1))
```

$t0=t_0$ 在表 14.5 中第 4 行第 4 列定义,$ci(1)=l$ 和 $ci(2)=u$ 分别为置信下限和置信上限。对 ttest2 的调用语句如下所述:

```
[x1,x2] = DataFci;
[h,p,ci] = ttest2(x1,x2,0.05)
```

结果为 $h=0$,即不能放弃原假设;$p=0.6445$, $ci(1)=-0.7550$ 和 $ci(2)=1.1855$ 分别为均差的置信下限和置信上限。由 p 值可以看出仅有 35.55% [即 100(1-0.6445)]的置信度可以相信平均值间有显著性差异,远远低于期望的置信度 95%,因此原假设不能放弃。

情况 7:

在情况 7 中仍以函数 **DataFci** 定义的表 14.4 中所列数据为例。在置信度为 95% 时确定这些样本的方差间是否存在显著性差异。假设:

$$H_0: \sigma_1^2 = \sigma_2^2$$

$$H_1: \sigma_1^2 \neq \sigma_2^2$$

检验统计为:

$$f_0 = \frac{s_1^2}{s_2^2}$$

放弃原假设的依据为:

$$f_0 > f_{\alpha/2,n_1-1,n_2-1} \text{ 或 } f_0 < f_{1-\alpha/2,n_1-1,n_2-1}$$

执行下面的程序:

```
[x1,x2] = DataFci;
L1 = length(x1);
```

```
L2 = length(x2);
ratio = var(x1)/var(x2);
if ratio > finv(0.975,L1 - 1,L2 - 1)
  disp('Reject null hypothesis')
  disp(['pValue = 'num2str(2 * (1 - fcdf(ratio,L1 - 1,L2 - 2)))])
elseif ratio < finv(0.025,L1 - 1,L2 - 1)
  disp('Reject null hypothesis')
  disp(['pValue = 'num2str(2 * fcdf(ratio,L1 - 1,L2 - 1))])
else
  disp('Null hypothesis cannot be rejected')
end
```

结果如下：

```
Reject null hypothesis
pValue = 6.5379e-005
```

由上述结果可以看出，样本方差间有显著性差异。值得注意的是对一般双限置信区间其 F 检验的 p 值由以下语句求得：

```
p = 2 * (1 - fcdf(r,n1,n2))
```

其中，$r = s_1^2/s_2^2$，$n1 = n_1$ 和 $n2 = n_2$ 为对应的自由度。

14.5 线性回归

14.5.1 一元线性回归

回归分析统计是在分析两变量或更多变量间的关系并确定其模型时采用的技术。一元线性回归模型只有一个独立变量。假设过程的输入量为 x，输出响应为 y，则其线性模型为：

$$y = \beta_1 x + \beta_0$$

如果有 n 个独立变量 x_i，就应该有 n 个对应的输出响应 y_i，$(i = 1,2,\cdots,n)$，则 y 的估计值由下式确定：

$$\hat{y} = \hat{y}(x) = \hat{\beta}_1 x + \hat{\beta}_0 \qquad x_{min} \leqslant x \leqslant x_{max} \tag{14.23}$$

其中，x_{min} 为 x_i 的最小值，x_{max} 为 x_i 的最大值，$\hat{\beta}_1$ 和 $\hat{\beta}_0$ 分别为 β_1 和 β_0 的估计值，且由下式给定：

$$\begin{aligned}\hat{\beta}_1 &= \frac{S_{xy}}{S_{xx}} \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}\end{aligned} \tag{14.24}$$

式中，

$$\begin{aligned}\bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i \qquad & \bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ S_{xx} &= \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 \qquad & S_{xy} &= \sum_{i=1}^{n} x_i y_i - n\overline{xy}\end{aligned} \tag{14.25}$$

$\hat{\beta}_1$ 和 $\hat{\beta}_0$ 的值由函数 polyfit 求得(参照 5.4.2 节)，因此，

```
[c,s] = polyfit(x,y,1)
```

其中,$c(1)=\hat{\beta}_1$,$c(2)=\hat{\beta}_0$,s 为函数 polyconf 所需要的参数,其用法将在下面解释。

对于 $x_{mim}\leqslant x\leqslant x_{max}$,$y(x)$估计值的 $100(1-\alpha)\%$ 置信上下限为:

$$\hat{y}(x)-w(x)\leqslant y(x)\leqslant \hat{y}(x)+w(x) \tag{14.26}$$

式中,

$$w(x)=t_{\alpha/2,n-2}\hat{\sigma}\sqrt{1+\frac{1}{n}+\frac{(x-\bar{x})^2}{S_{xx}}}$$

$$\hat{\sigma}^2=\frac{SS_E}{n-2} \qquad SS_E=S_{yy}-\hat{\beta}_1 S_{xy}$$

$$S_{yy}=\sum_{i=1}^{n}y_i^2-n\bar{y}^2 \tag{14.27}$$

$w(x)$和 $\hat{y}(x)$由函数 polyconf 求得,如下所述:

```
[c,s] = polyfit(x,y,1)
[yhat,w] = polyconf(c,x,s,alpha)
```

其中,$yhat=\hat{y}(x)$,$w=w(x)$,$alpha=\alpha$,向量 x 确定 *yhat* 和 w 的取值。

检验由式(14.23)给出的回归模型是否恰当的方法是求其残差,即:

$$e_i=y_i-\hat{y}(x_i) \qquad i=1,2,\cdots,n \tag{14.28}$$

如果 e_i 近似为正态分布,则选用该模型是合适的。

另一种检验模型是否恰当的方法是引入可决系数 R^2,R^2 定义为:

$$R^2=1-\frac{SS_E}{S_{yy}} \tag{14.29}$$

值 $100R^2$ 为模型所描述的数据变化率的百分比。该值越接近 100%,模型越有效。R 称为相关系数。

下面介绍这些关系式的应用。

例 14.9 回归分析

有一组数据如表 14.6 所列,该数据由函数 **DataRegress1** 定义。注意表中的数据是无序的,由于在图上将这些无序数据与一条直线相连很麻烦,所以先对其按升序进行排列。然而在 polyfit 和 polyconf 函数中并不需要再次排序。因此,

```
function[x,y] = DataRegress1
xx = [2.38 2.44 2.70 2.98 3.32 3.12 2.14 2.86 3.50 3.20 2.78 2.70 2.36 2.42...
    2.62 2.80 2.92 3.04 3.26 2.30];
yy = [51.11 50.63 51.82 52.97 54.47 53.33 49.90 51.99 55.81 52.93 52.87 52.36...
     51.38 50.87 51.02 51.29 52.73 52.81 53.59 49.77];
[x,index] = sort(xx);
y = yy(index);
```

以上代码中,*index* 为 x 中每个元素排序前的位置,为保留 x 和 y 的对应关系必须记录 *index* 值。如果对 x 和 y 都用 sort 函数排序,则其对应关系将被破坏。

表 14.6　用于一元线性回归分析的数据——DataRegress1

x	y	x	y
2.38	51.11	2.78	52.87
2.44	50.63	2.70	52.36
2.70	51.82	2.36	51.38
2.98	52.97	2.42	50.87
3.32	54.47	2.62	51.02
3.12	53.33	2.80	51.29
2.14	49.90	2.92	52.73
2.86	51.99	3.04	52.81
3.50	55.81	3.26	53.59
3.20	52.93	2.30	49.77

下面给出程序，求 $\hat{\beta}_1$ 和 $\hat{\beta}_0$，并在图上画出 $\hat{y}(x)$ 及置信度为 95% 的置信区间。在图中画出各个数据点，然后将这些点与直线相连，再加上恰当的注释。此外，还计算出可决系数并标于图中。运行结果参见图 14.11。

```
[x,y] = DataRegress1;
[c,s] = polyfit(x,y,1);
[yhat,w] = polyconf(c,x,s,0.05);
syy = sum(y.^2) - length(x) * mean(y)^2;
sse = syy - c(1) * (sum(x. * y) - length(x) * mean(x) * mean(y));
plot(x,yhat,'k-',x,yhat-w,'k--',x,yhat + w,'k--',x,y,'ks',[x;x],[yhat;y],'k-')
legend('Regression line','95% confidence interval of y','Location','SouthEast')
axis([2,3.6,48,57])
xlabel('x(Input)')
ylabel('y(Response)')
text(2.1,56,['Coefficient of determination R^2 = 'num2str(1 - sse/syy,3)])
```

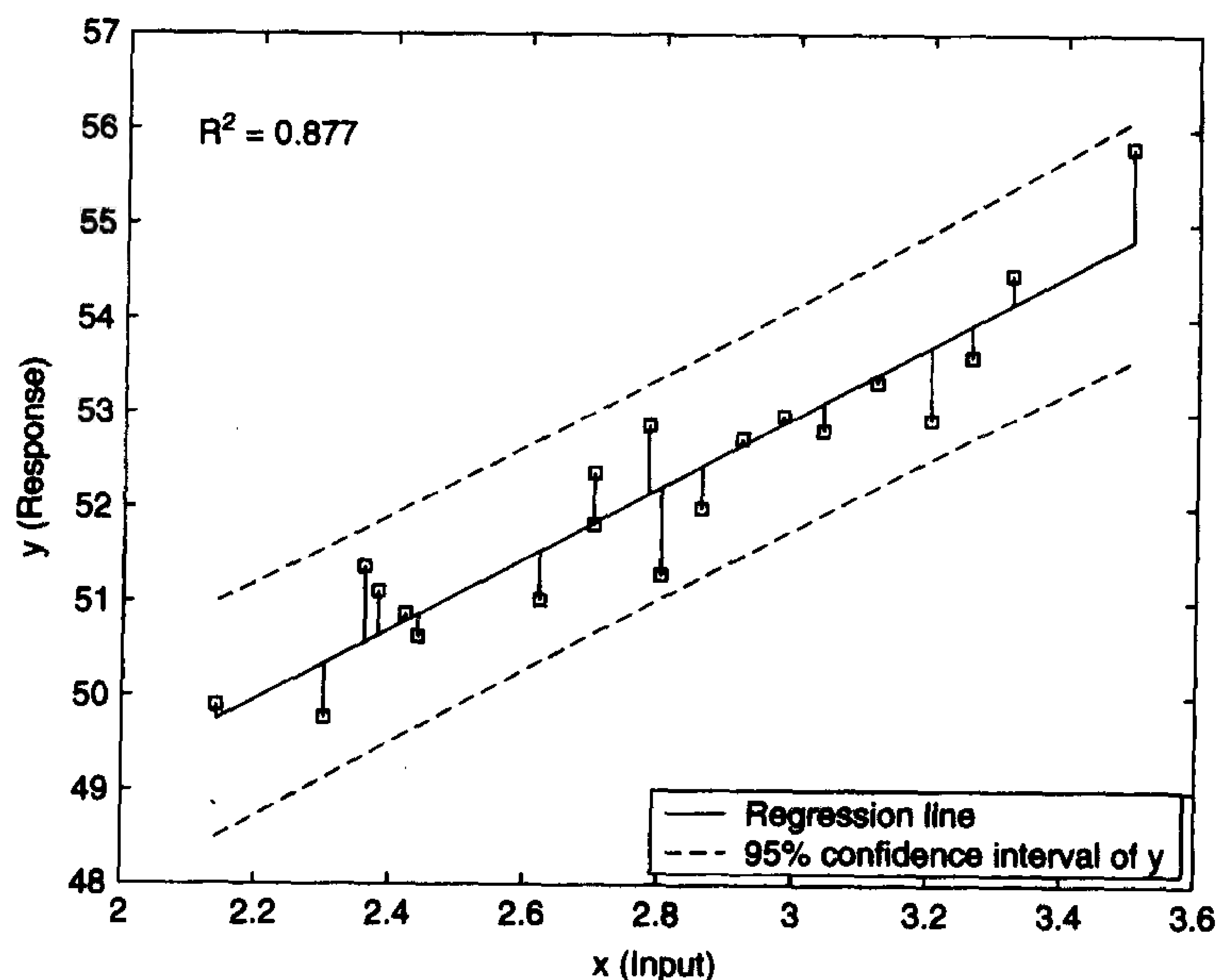


图 14.11　对表 14.6 所列数据进行线性回归分析的结果及 y 的置信上限和置信下限图

进一步分析残差。首先,计算残差值然后用函数 normplot(参见图 14.7)将其画在图中,以观察残差是否服从正态分布。程序为:

```
[x,y] = DataRegress1;
normplot(y-ployval(polyfit(x,y,1),x))
```

执行结果如图 14.12 所示。由图中看出这些残差点与直线非常接近,由此得出结论,这些残差值非常接近于正态分布,选择的模型是合适的。

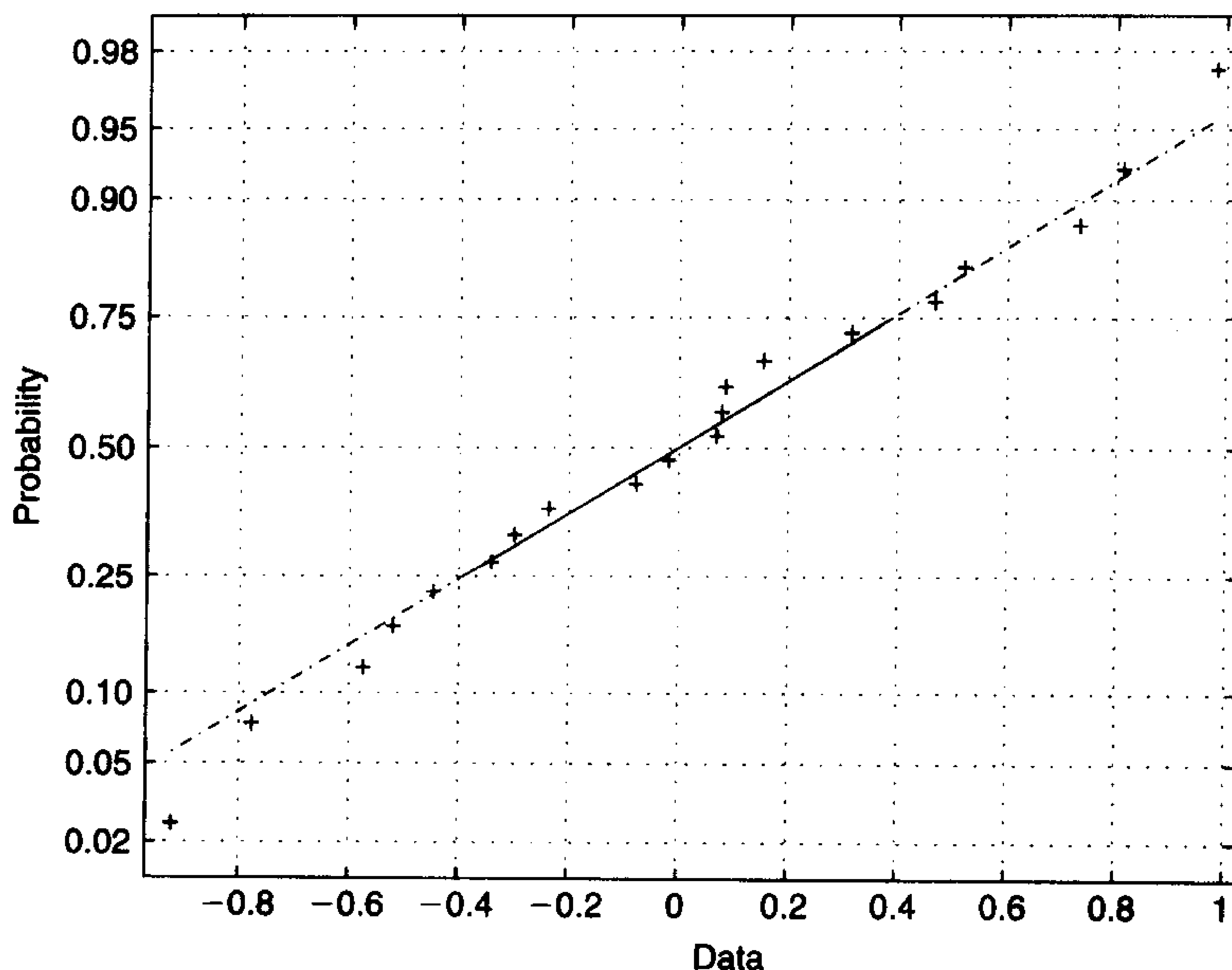


图 14.12　在图 14.11 所示直线附近的残差值的正态累积分布图

14.5.2　多元线性回归

实际应用中,可能影响过程输出量的因素(变量)多于一个,在这种情况下,需要建立多元回归模型。

在一个过程中,有一个输出 y 和 k 个输入 $x_j, j=1,2,\cdots,k$,这个过程可用以下述模型描述:

$$y = \beta_o + \sum_{j=1}^{k} \beta_j x_j \tag{14.30}$$

上式为具有 k 个独立变量的多元线性回归模型。$\beta_j, j=0,1,2,\cdots,k$,称做回归系数。更复杂的问题也可以用多元线性回归模型描述。例如,在三次多项式中只有一个独立变量 x,如下所述:

$$y = \beta_o + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

如果令 $x_1 = x, x_2 = x^2, x_3 = x^3$,则得到式(14.30)所示的线性模型,即:

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

而这类模型用函数 polyfit 更易求解。

另一个例子为:

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

则该方程将与式(14.30)描述 $x_3=x_1^2, x_4=x_2^2, x_5=x_1x_2$ 的形式一致。

由此看出，任何回归模型当与参数 β_j 呈线性关系时，无论其输出 y 的曲面形状如何，都是一个线性回归模型。

为了估计一些参数值，可以做 n 次重复试验($n>k+1$)。对应于每组数据 $x_{ij}, i=1,2,\cdots,n$ 及 $j=1,2,\cdots,k$，通过实验得到一组输出 y_i。用表格形式表示为：

y	x_1	x_2	$\cdots$	x_k
y_1	x_{11}	x_{12}	$\cdots$	x_{1k}
y_2	x_{21}	x_{22}	$\cdots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	x_{n1}	x_{n2}	$\cdots$	x_{nk}

式(14.30)变为下列形式：

$$y_i = \beta_o + \sum_{j=1}^{k} \beta_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.31}$$

如果数据用矩阵表示，则：

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} \tag{14.32}$$

β_j 的估计值用 $\hat{\beta}_j$ 表示，可由以下矩阵方程求得：

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{14.33}$$

矩阵 $\boldsymbol{X}$ 一般不是方阵。则，

$$\hat{y}_i = \hat{\beta}_o + \sum_{j=1}^{k} \hat{\beta}_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.34}$$

式中，$\hat{y}_i$ 为 y_i 的估计值。

求得回归系数后，还需要计算残差值以分析是否服从正态分布，作为验证模型正确性的一个指标。残差定义如下：

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$$

则，

$$e_i = y_i - \hat{y}_i = y_i - \hat{\beta}_o - \sum_{j=1}^{k} \hat{\beta}_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.35}$$

回归系数 β_j 的置信区间如下：

$$\beta_{Lj} \leqslant \beta_j \leqslant \beta_{Uj} \qquad j = 0,1,\cdots,k \tag{14.36}$$

其中，

$$\begin{aligned} \beta_{Lj} &= \hat{\beta}_j - t_{\alpha/2,n-k-1}\hat{\sigma}\sqrt{C_{jj}} \\ \beta_{Uj} &= \hat{\beta}_j + t_{\alpha/2,n-k-1}\hat{\sigma}\sqrt{C_{jj}} \end{aligned} \tag{14.37}$$

和

$$\hat{\sigma}^2 = \frac{y'y - \hat{\beta}'X'y}{n-k-1}$$

$$C = (X'X)^{-1} = \begin{bmatrix} C_{00} & C_{01} & \cdots & C_{0k} \\ C_{10} & C_{11} & & \\ \vdots & & \ddots & \\ C_{k0} & & & C_{kk} \end{bmatrix} \tag{14.38}$$

换句话说,

$$\text{var}(\hat{\beta}_j) = \hat{\sigma}^2 C_{jj} \quad j = 0,1,\cdots,k$$

为 $\hat{\beta}_j$ 的方差估计值,及

$$\text{covar}(\hat{\beta}_i, \hat{\beta}_j) = \hat{\sigma}^2 C_{ij} \quad i,j = 0,1,\cdots,k \quad i \neq j$$

是 $\hat{\beta}_i$ 和 $\hat{\beta}_j$ 的协方差估计值。

多元可决系数 R^2 为:

$$R^2 = 1 - \frac{y'y - \hat{\beta}'X'y}{y'y - n\bar{y}^2} \tag{14.39}$$

其中,

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$$

R 为相关系数。

下面进行假设检验,检验是否至少有一个回归变量($x_i, i = 1,2,\cdots,k$)与输出响应 y 之间存在线性关系。假设检验为:

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$$
$$H_1: \beta_j \neq 0 \quad (\text{至少有一个 } j \text{ 值})$$

放弃 H_0 假设意味着至少有一个回归变量对统计做出显著贡献。检验统计量定义为:

$$F_0 = \frac{(\hat{\beta}'X'y - n\bar{y}^2)/k}{(y'y - \hat{\beta}'X'y)/(n-k-1)} \quad n > k+1 \tag{14.40}$$

如果,

$$F_0 > f_{\alpha,k,n-k-1}$$

则放弃 H_0。

对以上方程的数值计算可通过语句:

```
beta = regress(y,x)
```

或语句

```
[beta,betacl,e,ecl,stats] = regress(y,X,alpha)
```

实现。

其中,

beta = [$\hat{\beta}_0$ $\hat{\beta}_1 \cdots \hat{\beta}_k$]在式(14.33)中定义

betacl 为 (($k+1$) × 2)维数组,存放置信下限 β_L 和置信上限 β_U 值,如式(14.37)所定义的那样。阶数由 *beta* 确定。

$e=[e_1\ e_2\cdots e_n]$为式(14.35)给出的残差；

ecl 为残差的置信区间

stats $=[R^2\ F_0\ p]$，其中，

R^2 由式 (14.39)给出

F_0 由式(14.40)给出

p 为 F_0 对应的 p 值，即：

p = 1 - fcdf(F0,k,n-k-1)

$y=[y_1\ y_2\cdots y_n]'$为输出响应列向量

$\boldsymbol{X}$ 等于由式(14.32)定义的 $\boldsymbol{X}$

alpha $=\alpha$

下面给出这些公式的具体用法。

例 14.10 多元回归分析

有一组数据如表 14.7 所示，现用以下方程模拟这些数据：

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\beta_4x_1x_2+\beta_5x_1x_3+\beta_6x_2x_3+\beta_7x_1^2+\beta_8x_2^2+\beta_9x_3^2$$

首先，创建函数 **DataMultiRegress1**，根据式(14.32)生成 $\boldsymbol{X}$ 变量：

```
function[y,X] = DataMultiRegress1
y = [0.22200 0.39500 0.42200 0.43700 0.42800 0.46700 0.44400 0.37800 0.49400 ...
   0.45600 0.45200 0.11200 0.43200 0.10100 0.23200 0.30600 0.09230 0.11600 ...
   0.07640 0.43900 0.09440 0.11700 0.07260 0.04120 0.25100 0.00002]';
x1 = [7.3 8.7 8.8 8.1 9.0 8.7 9.3 7.6 10.0 8.4 9.3 7.7 9.8 7.3 8.5 9.5 7.4 7.8 7.7 10.3 ...
   7.8 7.1 7.7 7.4 7.3 7.6]';
x2 = [0.0 0.0 0.7 4.0 0.5 1.5 2.1 5.1 0.0 3.7 3.6 2.8 4.2 2.5 2.0 2.5 2.8 2.8 3.0 1.7 ...
   3.3 3.9 4.3 6.0 2.0 7.8]';
x3 = [0.0 0.3 1.0 0.2 1.0 2.8 1.0 3.4 0.3 4.1 2.0 7.1 2.0 6.8 6.6 5.0 7.8 7.7 8.0 ...
   4.2 8.5 6.6 9.5 10.9 5.2 20.7]';
X = [ones(length(y),1)x1,x2,x3,x1.*x2,x1.*x3,x2.*x3,x1.^2 x2.^2,x3.^2];
```

表 14.7 函数 DataMultiRegress1 定义的数据表

y	x_1	x_2	x_3	y	x_1	x_2	x_3
0.222 00	7.3	0.0	0.0	0.101 00	7.3	2.5	6.8
0.395 00	8.7	0.0	0.3	0.232 00	8.5	2.0	6.6
0.422 00	8.8	0.7	1.0	0.306 00	9.5	2.5	5.0
0.437 00	8.1	4.0	0.2	0.092 30	7.4	2.8	7.8
0.428 00	9.0	0.5	1.0	0.116 00	7.8	2.8	7.7
0.467 00	8.7	1.5	2.8	0.076 40	7.7	3.0	8.0
0.444 00	9.3	2.1	1.0	0.439 00	10.3	1.7	4.2
0.378 00	7.6	5.1	3.4	0.094 40	7.8	3.3	8.5
0.494 00	10.0	0.0	0.3	0.117 00	7.1	3.9	6.6
0.456 00	8.4	3.7	4.1	0.072 60	7.7	4.3	9.5
0.452 00	9.3	3.6	2.0	0.041 20	7.4	6.0	10.9
0.112 00	7.7	2.8	7.1	0.251 00	7.3	2.0	5.2
0.432 00	9.8	4.2	2.0	0.000 02	7.6	7.8	20.7

然后,计算出系数 $\hat{\beta}_j$ 的估计值及其在 95% 置信度上的置信区间。在窗口中显示 R^2, F_0 及其 p 值,并在图上绘出残差值,以确定残差是否服从正态分布。下面为程序:

```
[y,X] = DataMultiRegress1;
[b,bcl,e,ecl,stat] = regress(y,X,0.05);
lenb = length(b);
disp('Regression coefficients and their confidence limits')
disp([num2str(bcl(:,1))repmat('<= beta(',lenb,1)num2str((0:lenb-1)')...
  repmat(') = ',lenb,1)num2str(b)repmat('<= ',lenb,1)num2str(bcl(:,2))])
disp(['Coefficient of determination R^2 = 'num2str(stat(1))])
disp(['Test statistic F0 = 'num2str(stat(2))'and corresponding p-value = '...
         num2str(stat(3))])
normplot(e)
```

执行结果在 MATLAB 命令窗口中显示出下列数据。并在图中列出了这些数据,如图14.13所示。从图中可以看出,除有 5 个点以外其余残差点都落在正态分布直线的附近,因此该模型是合适的。

```
Regression coefficients and their confidence limits
-4.4976      <= beta(0) = -1.7694      <= 0.9589
-0.20282     <= beta(1) = 0.4208       <= 1.0444
-0.054708    <= beta(2) = 0.22245      <= 0.49961
-0.27691     <= beta(3) = -0.128       <= 0.020918
-0.045395    <= beta(4) = -0.019876    <= 0.0056419
-0.0070049   <= beta(5) = 0.0091515    <= 0.025308
-0.012346    <= beta(6) = 0.0025762    <= 0.017499
-0.054932    <= beta(7) = -0.019325    <= 0.016283
-0.032989    <= beta(8) = -0.0074485   <= 0.018092
-0.002231    <= beta(9) = 0.00082397   <= 0.003879
Coefficient of determination R^2 = 0.91695
Test statistic F0 = 19.628 and corresponding p-value = 5.0513e-007
```

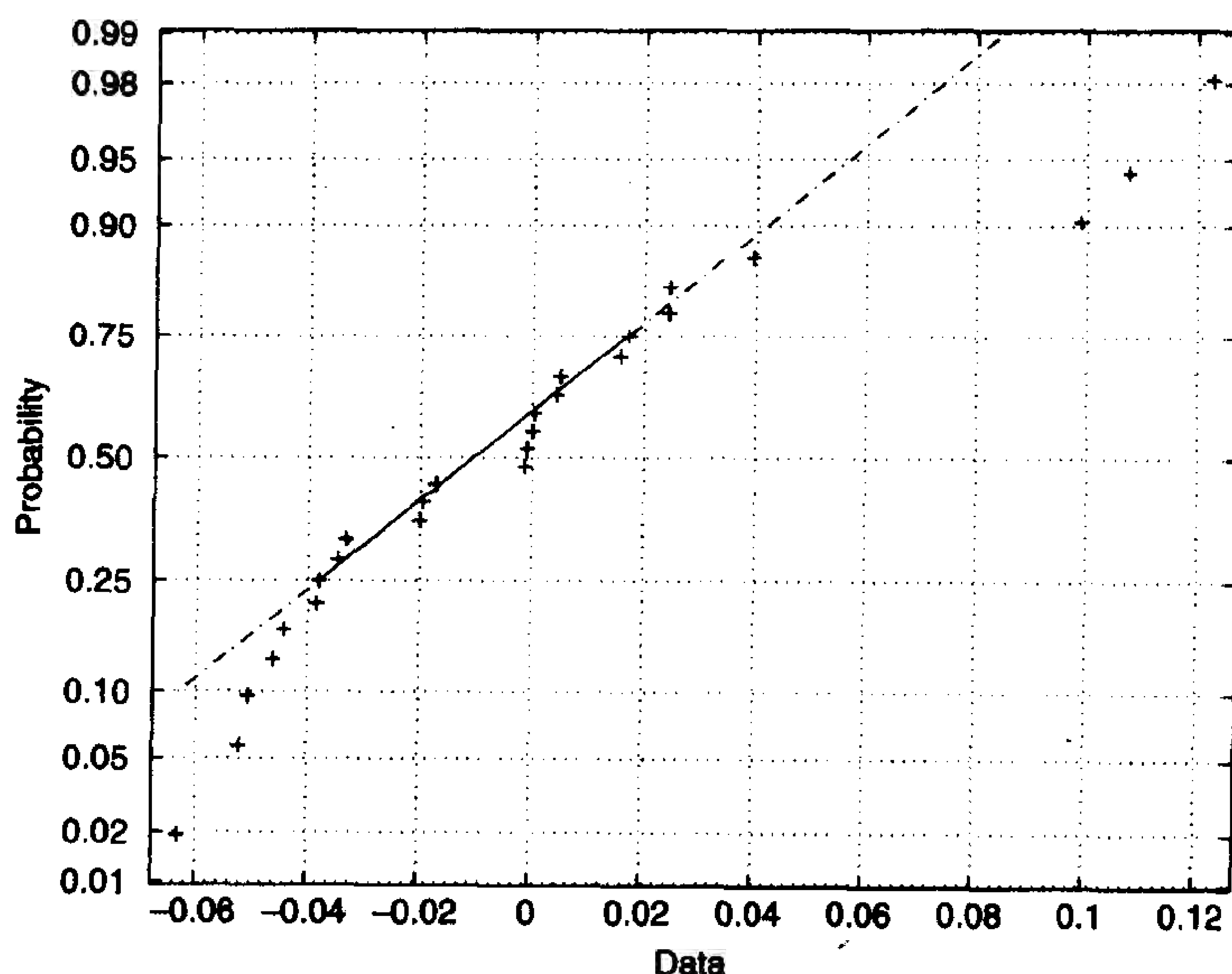


图 14.13 表 14.7 中数据的残差正态累积分布图

14.6　试验设计

14.6.1　单因素实验：方差分析

在单因素实验中，因素用 A 表示。在一次实验过程中，将 A 取不同的水平 $A_j, j=1,2,\cdots,a$，对同样的实验重复做 n 次即得到 n 次重复。结果数据填入表 14.8 中。在表 14.8 中，第 1 列数据是观察值 x_{j1}，是对水平 A_j，$(j=1,2,\cdots,a)$ 随机排序后测得的实验数据。第 2 列观察值 $x_{j2}, j=1,2,\cdots,a$，是对水平 A_j 重新随机排序后测得的实验数据，上述过程重复做 n 次可得到 n 组不同数据。做实验时，要保证测出数据 x_{jk} 的独立性。可定义两个独立方差，定义中用到的参数 μ_i 和 s_i^2，参见表 14.8。因素 A 的平均方差为：

$$s_A^2 = \frac{n}{a-1}\left(\sum_{i=1}^{a}\mu_i^2 - a\bar{x}^2\right) = \frac{SS_A}{a-1}$$

上式中，自由度为 $a-1$，$\bar{x}$ 为总平均值，定义如下：

$$\bar{x} = \frac{1}{an}\sum_{i=1}^{a}\sum_{j=1}^{n}x_{ij}$$

误差的方差为：

$$s_{\text{error}}^2 = \frac{1}{a}\sum_{i=1}^{a}s_i^2 = \frac{1}{a(n-1)}\left(\sum_{i=1}^{a}\sum_{j=1}^{n}x_{ij}^2 - an\bar{x}^2\right) = \frac{SS_{\text{error}}}{a(n-1)}$$

其中，$a(n-1)$ 为自由度。

表 14.8　重复 $n>1$ 次的单因素实验表

水平	观察值				均值	方差	残差
A_1	x_{11}	x_{12}	$\cdots$	x_{1n}	$\mu_1 = \frac{1}{n}\sum_{j=1}^{n}x_{1j}$	$s_1^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_{1j}-\mu_1)^2$	$\varepsilon_{1j} = x_{1j} - \mu_1$
A_2	x_{21}	x_{22}		x_{2n}	$\mu_2 = \frac{1}{n}\sum_{j=1}^{n}x_{2j}$	$s_2^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_{2j}-\mu_2)^2$	$\varepsilon_{2j} = x_{2j} - \mu_2$
$\cdots$				$\cdots$	$\cdots$	$\cdots$	
A_a	x_{a1}	x_{a2}	$\cdots$	x_{an}	$\mu_a = \frac{1}{n}\sum_{j=1}^{n}x_{aj}$	$s_a^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_{aj}-\mu_a)^2$	$\varepsilon_{aj} = x_{aj} - \mu_a$

下面的等式将两个方差 s_A^2 和 s_{error}^2 联系起来：

$$\begin{aligned} SS_{\text{total}} &= \sum_{i=1}^{a}\sum_{j=1}^{n}(x_{ij}-\bar{x})^2 = \sum_{i=1}^{a}\sum_{j=1}^{n}[(\mu_i-\bar{x})+(x_{ij}-\mu_i)]^2 \\ &= n\sum_{i=1}^{a}(\mu_i-\bar{x})^2 + \sum_{i=1}^{a}\sum_{j=1}^{n}(x_{ij}-\mu_i)^2 \\ &= SS_A + SS_{\text{error}} \\ &= (a-1)s_A^2 + a(n-1)s_{\text{error}}^2 \end{aligned}$$

等式左边称为总平方和，其自由度为 $an-1$。等式将总方差分为两个独立的部分：一部分与因素 A 有关，另一部分与过程数据的误差有关，该残差用 ε_{ij} 表示。

方差分析(ANOVA)中，习惯上定义一个均方值参数，用 MS 表示。MS 等于平方和除以自由度，因此对单因素实验为：

$$MS_A = \frac{SS_A}{(a-1)} = s_A^2 \qquad (a>1)$$

$$MS_{error} = \frac{SS_{error}}{a(n-1)} = s^2_{error} \quad (n > 1)$$

实验目的是要确定因素 A 在不同水平上对输出 x_{ij} 是否有显著的影响。为此,定义因素 A 的均方值与独立随机误差均方值的比。由比值大小可知 A 的方差在总方差中是否占有显著位置。因此,检验统计量定义为:

$$F_0 = \frac{MS_A}{MS_{error}}$$

建立假设为:

$$H_0: \mu_1 = \mu_2 = \ldots = \mu_a$$

$$H_1: \mu_j \neq \mu_i \text{ 至少有一个 } j \neq i$$

若,

$$F_0 > f_{\alpha, a-1, a(n-1)}$$

则原假设没有被放弃。分析结果通常以表格的形式给出,如表 14.9 所示。

表 14.9　重复 $n>1$ 次的单因素方差分析(ANOVA)表

因素	平方和	自由度	均方值	F_0	p 值
A	SS_A	$a-1$	MS_A	MS_A/MS_{error}	
Error	SS_{error}	$a(n-1)$	MS_{error}		
Total	SS_{total}	$an-1$			

单因素方差分析可由

```
p = anova1(x)
```

实现。其中,p 为 p 值,x 为对表 14.8 的列数据转换后的数据。该函数另外有两个输出:一个为表 14.9 所示的方差分析(ANOVA)表,另一个是对应 a 水平的中位数变化盒状图。下面举例说明单因素实验的方差分析。

例 14.11　单因素方差分析(ANOVA)

有一组数据如表 14.10 所示。下面编写程序绘出方差分析 ANOVA 表,显示 p 值,计算残差并分析残差是否服从正态分布。首先,创建一函数 **DataAnova1**,将数据转换为函数 anova1 能接受的形式,如下所示:

```
function d = DataAnova1
d = [143 141 150 146;...
    152 149 137 143;...
    134 133 132 127;...
    129 127 132 129;...
    147 148 144 142]';
```

表 14.10　例 14.11 中的数据——DataAnova1

水平	观察值			
1	143	141	150	146
2	152	149	137	143
3	134	133	132	127

（续表）

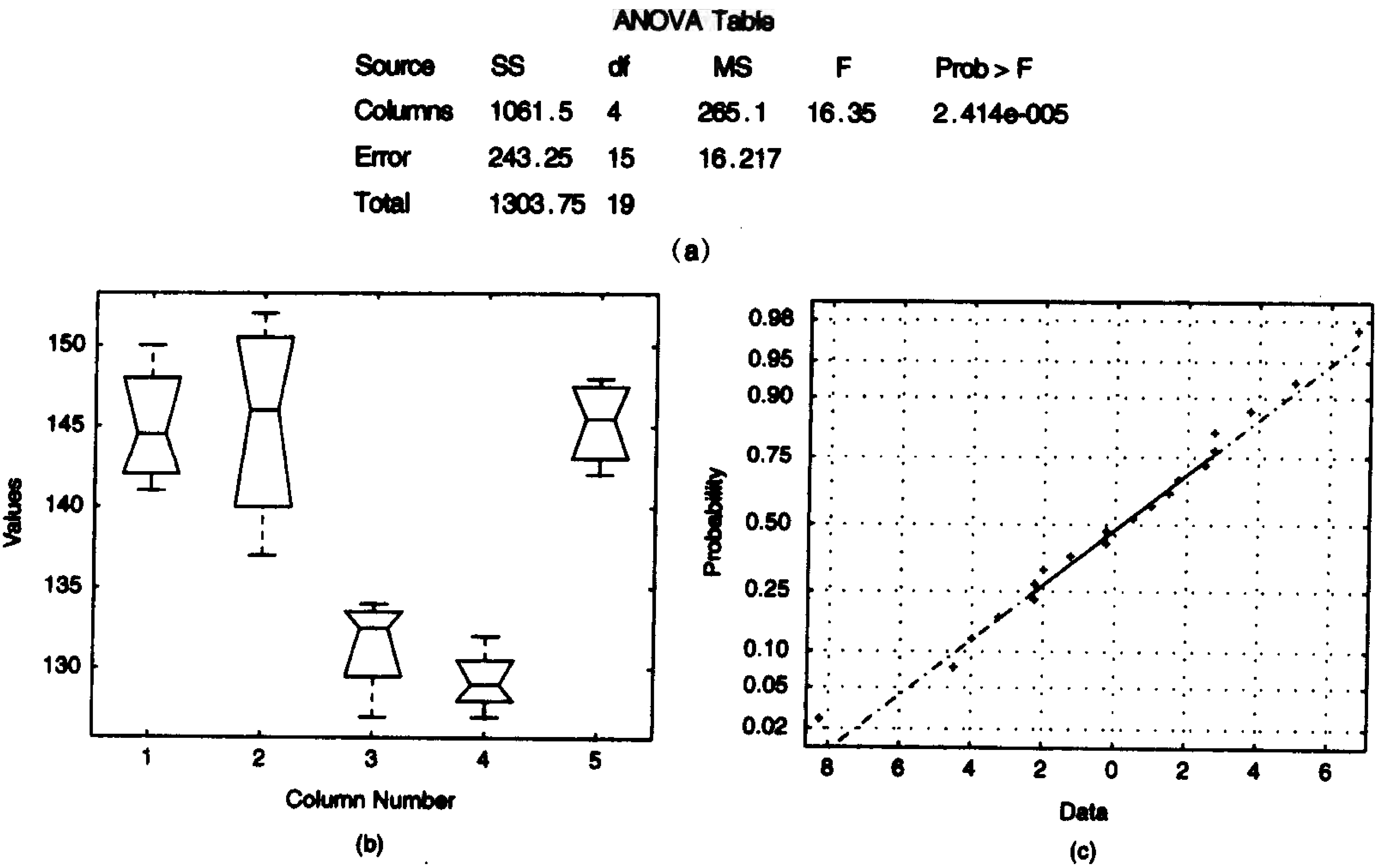
水平	观察值			
4	129	127	132	129
5	147	148	144	142

程序如下所示：

```
vv = DataAnova 1;
[r,c] = size(vv);
pp = anova1(vv);
meen = mean(vv);
k = 0;
for n = 1:r
 for m = 1:c
  k = k + 1;
  e(k) = vv(n,m)-meen(m);
 end
end
figure
normplot(e)
```

上述代码中在 for 循环内将所有残差放于一阶向量中。函数 figure 用于打开另一图形窗口。因为 anova1 函数自己会打开两个窗口，如果不使用 figure 函数，则由函数 anova1 生成的两个图形窗口的其中一个将被重写。

anova1 函数创建的表格如图 14.14(a)所示，生成的盒状图如图 14.14(b)所示。图 14.14(c)是由 normplot 函数生成的。



ANOVA Table

Source	SS	df	MS	F	Prob > F
Columns	1061.5	4	265.1	16.35	2.414e-005
Error	243.25	15	16.217		
Total	1303.75	19			

(a)

图 14.14　表 14.10 中数据的方差分析：(a) ANOVA 表；(b)5种水平的盒状图；(c)残差的正态分布图

14.6.2 多因素实验

析因实验

单因素实验结果可延伸用于多因素实验,后者称做析因实验。析因实验中每次重复要包括每个因素的所有水平的各种不同组合情况。现用两因素实验说明这一点。设实验因素 A 具有 a 个水平,因素 B 具有 b 个水平,实验重复次数为 $n(n>1)$,输出为 x_{ijk},其中 $i=1,2,\cdots,a;j=1,2,\cdots,b;k=1,2,\cdots,n$。每个因素的不同水平间的间隔不一定相等。实验数据如表14.11所示。

表 14.11 两因素析因实验中的数据

		因素 B			
		1	2	…	b
因素 A	1	$y_{111},y_{112},\cdots,y_{11n}$	$y_{121},y_{122},\cdots,y_{12n}$		$y_{1b1},y_{1b2},\cdots,y_{1bn}$
	2	$y_{211},y_{212},\cdots,y_{21n}$	$y_{221},y_{222},\cdots,y_{22n}$		$y_{2b1},y_{2b2},\cdots,y_{2bn}$
	⋮				
	a	$y_{a11},y_{a12},\cdots,y_{a1n}$	$y_{a21},y_{a22},\cdots,y_{a2n}$		$y_{ab1},y_{ab2},\cdots,y_{abn}$

首先,应从平方和等式分析入手。但在此之前,先引入下列几个不同平均值的定义:

$$\bar{x}_{ijn}=\frac{1}{n}\sum_{k=1}^{n}x_{ijk}$$

$$\bar{x}_{ibn}=\frac{1}{b}\sum_{j=1}^{b}\bar{x}_{ijn}=\frac{1}{bn}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}$$

$$\bar{x}_{ajn}=\frac{1}{a}\sum_{i=1}^{a}\bar{x}_{ijn}=\frac{1}{an}\sum_{i=1}^{a}\sum_{k=1}^{n}x_{ijk}$$

总平均值为:

$$\bar{x}=\frac{1}{abn}\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}$$

两因素方差分析的总平方和等式为:

$$SS_{total}=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(x_{ijk}-\bar{x})^2=SS_A+SS_B+SS_{AB}+SS_{error}$$

其中,

$$SS_A=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{x}_{ibn}-\bar{x})^2=bn\sum_{i=1}^{a}\bar{x}_{ibn}^2-abn\bar{x}^2$$

$$SS_B=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{x}_{ajn}-\bar{x})^2=an\sum_{j=1}^{b}\bar{x}_{ajn}^2-abn\bar{x}^2$$

$$SS_{AB}=n\sum_{i=1}^{a}\sum_{j=1}^{b}(\bar{x}_{ijn}-\bar{x}_{ibn}-\bar{x}_{ajn}+\bar{x})^2$$

和

$$SS_{error}=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(x_{ijk}-\bar{x}_{ijn})^2=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}^2-n\sum_{i=1}^{a}\sum_{j=1}^{b}\bar{x}_{ijn}^2$$

SS_A,SS_B,SS_{AB},SS_{error}及SS_{total}的自由度分别为$(a-1)$,$(b-1)$,$(a-1)(b-1)$,$ab(n-1)$及

($abn-1$)。平方和 SS_{AB} 为因素 A 和 B 的交互效应的平方和。两因素实验的方差分析表如表 14.12所示，表中还给出了均方值的定义。可以看出，方差分析中将两因素的交互效应单独分离出来，通过求 MS_{AB}/MS_{error} 来判断两因素交互效应在给定置信度下统计意义上的显著性。

表 14.12 重复 $n>1$ 次的两因素实验方差分析(ANOVA)表

因素	平方和	自由度	均方值	F_0	$f_{a,z,ab(n-1)}$	p 值
A	SS_A	$a-1$	$MS_A=SS_A/(a-1)$	MS_A/MS_{error}	(f 表, $z=a-1$)	
B	SS_B	$b-1$	$MS_B=SS_B/(b-1)$	MS_B/MS_{error}	(f 表, $z=b-1$)	
AB	SS_{AB}	$(a-1)(b-1)$	$MS_{AB}=SS_{AB}/(a-1)(b-1)$	MS_{AB}/MS_{error}	(f 表, $z=(a-1)(b-1)$)	
Error	SS_{error}	$ab(n-1)$	$MS_{error}=SS_{error}/ab(n-1)$			
Total	SS_{total}	$abn-1$				

两因素析因实验的求解可由语句

```
anova2(y,n)
```

实现。其中，n 为重复次数，函数产生表 14.12。表 14.11 中的数据用 $\boldsymbol{y}$ 矩阵表示如下：

$$\boldsymbol{y}=\begin{bmatrix} y_{111} & y_{121} & \cdots & y_{1b1} \\ y_{112} & y_{122} & & y_{1b2} \\ \vdots & & & \\ y_{11n} & y_{12n} & & y_{1bn} \\ y_{211} & y_{221} & & y_{2b1} \\ y_{212} & y_{222} & & y_{2b2} \\ \vdots & & & \\ y_{21n} & y_{22n} & & y_{2bn} \\ \vdots & & & \\ y_{a11} & y_{a21} & & y_{ab1} \\ y_{a12} & y_{a22} & & y_{ab2} \\ \vdots & & & \\ y_{a1n} & y_{a2n} & & y_{abn} \end{bmatrix}$$

下面举例说明这些关系。

例 14.12 两因素方差分析(ANOVA)

一组数据如表 14.13 所示，下面创建函数存放表中数据：

```
function d = DataAnova2
dc1 = [[130,155,74,180]';[150,188,159,126]';[138,110,168,160]'];
dc2 = [[34,40,80,75]';[136,122,106,115]';[174,120,150,139]'];
dc3 = [[20,70,82,58]';[25,70,58,45]';[96,104,82,60]'];
d = [dc1,dc2,dc3];
```

程序为：

```
anova2(DataAnova2,4);
```

执行结果在图形窗口中显示下列表格:

ANOVA Table

Source	SS	df	MS	F	Prob > F
Columns	3.912e+004	2	1.956e+004	28.97	0
Rows	1.068e+004	2	5342	7.911	0.002
Interaction	9614	4	2403	3.56	0.0186
Error	1.823e+004	27	675.2		
Total	7.765e+004	35			

由 p 值看出,因素 A 和因素 B 具有置信度高于 99.8% 的统计意义上的显著性,其交互效应有置信度 98% 的统计意义上的显著性。

表 14.13 例14.12 的数据——DataAnova2

		因素 B		
		1	2	3
因素 A	1	130,155,74,180	34,40,80,75	20,70,82,58
	2	150,188,159,126	136,122,106,115	25,70,58,45
	3	138,110,168,160	174,120,150,139	96,104,82,60

2^K 析因实验

上节描述的析因实验中如果包含 k 个因素且每个因素只有两个水平,则此类析因实验称做 2^k 析因实验。2^k 析因实验假定在每个因素的两个水平之间存在着线性关系,这种假定给怎样进行实验及分析实验带来了一些简化。

习惯上某因素的高水平用"1"或"+"表示,其低水平用"0"或"-"表示。表 14.14 中列出 k 个因素($k=2,3,4$),取不同水平的 2^k 个因素组合也在表 14.14 中给出,一次执行顺序即为一次重复。下面介绍表的用法:对于 2^2($k=2$)析因实验,只有标为 A 和 B 的两列及前 4 行数据有效($m=1,\cdots,4$)。两个因素的四种组合按随机顺序组成,标为 2^2 的一列即为一种随机组合。首先执行第 2 行的组合:A 为高水平(A_{high}),B 为低水平(B_{low}),其输出值为 $y_{2,1}$。然后是第 4 行的组合,其中 A 和 B 都为高水平(分别为 A_{high}和 B_{high}),其输出值为 $y_{4,1}$。其余两种组合都做完后,就完成了实验的一次重复。如果重新获得的随机顺序不同于 2^2 列中的顺序,则在该顺序下做四种不同组合,得到第 2 次重复的输出。$k=3$ 时,因素为 A,B 和 C,表中前 8 行有效;$k=4$ 时,因素有 A,B,C 和 D,表中所有 16 行数据均有效。表中标为 2^3 和 2^4 的列分别是上述两种情况下的一种随机执行顺序。

表 14.14 不同次序下每个因素在不同水平下的 2^2,2^3 及 2^4 析因实验

运行序列号	因数及其水平				数据($y_{m,j}$)			运行次序号 *		
m	A	B	C	D	$j=1$	$j=2$	…	2^2	2^3	2^4
1	-	-	-	-	$y_{1,1}$	$y_{1,2}$		3	5	6
2	+	-	-	-	$y_{2,1}$	$y_{2,2}$		1	7	11
3	-	+	-	-	$y_{3,1}$	$y_{3,2}$		4	8	14

(续表)

运行序列号	因数及其水平				数据($y_{m,j}$)			运行次序号*		
m	A	B	C	D	$j=1$	$j=2$	…	2^2	2^3	2^4
4	+	+	–	–	$y_{4,1}$	$y_{4,2}$		2	4	5
5	–	–	+	–	$y_{5,1}$	$y_{5,2}$			2	13
6	+	–	+	–	$y_{6,1}$	$y_{6,2}$			1	2
7	–	+	+	–	$y_{7,1}$	$y_{7,2}$			3	16
8	+	+	+	–	$y_{8,1}$	$y_{8,2}$			6	15
9	–	–	–	+	$y_{9,1}$	$y_{9,2}$				9
10	+	–	–	+	$y_{10,1}$	$y_{10,2}$				7
11	–	+	–	+	$y_{11,1}$	$y_{11,2}$				10
12	+	+	–	+	$y_{12,1}$	$y_{12,2}$				3
13	–	–	+	+	$y_{13,1}$	$y_{13,2}$				8
14	+	–	+	+	$y_{14,1}$	$y_{14,2}$				4
15	–	+	+	+	$y_{15,1}$	$y_{15;2}$				1
16	+	+	+	+	$y_{16,1}$	$y_{16,2}$				12

* 只是 $j=1$ 的一种随机顺序,对 $j=2$ 将生成一种新的随机顺序,依次类推。

实验数据收集完整后,进行以下分析,设实验重复次数大于 1。观察表 14.15 中所列数据,列中符号"+"和"–"分别代表 +1 和 –1,列 A,B,C 和 D 与表 14.14 中的相同,表 14.14 中的符号"+"和"–"也分别代表 +1 和 –1。表示所有交互效应的列值由相应元素列值相乘得到。即标为 ABC 的列的符号值由列 A、列 B 和列 C 中的符号值相乘得到。例如,第 7 行($m=7$)的 $A=-1$,$B=+1$,$C=+1$,因此第 7 行标为 ABC 的列值为 $-1[(-1)(+1)(+1)]$。而且对 2^2 实验只用到表中前 3 列和前 4 行($m=1,\cdots,4$);对于 2^3 实验,用到表中前 7 列和前 8 行($m=1,\cdots,8$)数据;对于 2^4 实验,用到表中前 15 列和 16 行($m=1,\cdots,16$)数据。

表 14.15 在 2^2,2^3 及 2^4 实验中用于求平方和与均方值的不同量的定义

因素及其交互效应(λ)[①]															数据[②]			
A	B	AB	C	AC	BC	ABC	D	AD	BD	ABD	CD	ACD	BCD	$ABCD$	$j=1$	... $j=n$	$S_m^{\#}$	m
–	–	+	–	+	+	–	–	+	+	–	+	–	–	+	$y_{1,1}$	$y_{1,n}$	S_1	1
+	–	–	–	–	+	+	–	–	+	+	+	+	–	–	$y_{2,1}$	$y_{2,n}$	S_2	2
–	+	–	–	+	–	+	–	+	–	+	+	–	+	–	$y_{3,1}$	$y_{3,n}$	S_3	3
+	+	+	–	–	–	–	–	–	–	–	+	+	+	+	$y_{4,1}$	$y_{4,n}$	S_4	4
–	–	+	+	–	–	+	–	+	+	–	–	+	+	–	$y_{5,1}$	$y_{5,n}$	S_5	5
+	–	–	+	+	–	–	–	–	+	+	–	–	+	+	$y_{6,1}$	$y_{6,n}$	S_6	6
–	+	–	+	–	+	–	–	+	–	+	–	+	–	+	$y_{7,1}$	$y_{7,n}$	S_7	7

(续表)

因素及其交互效应(λ)①															数据②			
+	+	+	+	+	+	+	−	−	−	−	−	−	−	−	$y_{8,1}$	$y_{8,n}$	S_8	8
−	−	+	−	+	+	−	+	+	−	+	−	+	+	−	$y_{9,1}$	$y_{9,n}$	S_9	9
+	−	−	−	−	+	+	+	−	−	−	−	−	+	+	$y_{10,1}$	$y_{10,n}$	S_{10}	10
−	+	−	−	+	−	+	+	+	+	−	−	+	−	+	$y_{11,1}$	$y_{11,n}$	S_{11}	11
+	+	+	−	−	−	−	+	−	+	+	−	−	−	−	$y_{12,1}$	$y_{12,n}$	S_{12}	12
−	−	+	+	−	−	+	+	+	−	+	+	−	−	+	$y_{13,1}$	$y_{13,n}$	S_{13}	13
+	−	−	+	+	−	−	+	−	−	−	+	+	−	−	$y_{14,1}$	$y_{14,n}$	S_{14}	14
−	+	−	+	−	+	−	+	+	+	−	+	−	+	−	$y_{15,1}$	$y_{15,n}$	S_{15}	15
+	+	+	+	+	+	+	+	−	+	+	+	+	+	+	$y_{16,1}$	$y_{16,n}$	S_{16}	16

① “+”和“-”号分别表示+1和-1,也指因素的高水平和低水平。

② 数据的获取方法与表 14.14 一致。

* $S_m = \sum_{j=1}^{n} y_{m,j}$

具有 $n(n>1)$次重复 k 个因素的实验，其数据平方和为：

$$SS_{total} = \sum_{j=1}^{n}\sum_{m=1}^{2^k} y_m^2 - 2^k n\bar{y}^2$$

$$SS_{error} = SS_{total} - \sum_{\lambda} SS_\lambda$$

$$SS_\lambda = \frac{C_\lambda^2}{n2^k} \qquad \lambda = A, B, AB, \cdots$$

其中，

$$C_\lambda = \sum_{m=1}^{2^k} S_m \times (\lambda\text{ 列 } m \text{ 行中的符号}) \qquad \lambda = A, B, AB, \cdots$$

$$\bar{y} = \frac{1}{n2^k}\sum_{m=1}^{2^k} S_m$$

S_m 在表 14.15 中定义。

主因素及其交互效应产生的效果平均值由下式决定：

$$\text{Effect}_\lambda = \frac{C_\lambda}{n2^{k-1}} \qquad \lambda = A, B, AB, \cdots$$

式中，Effect_λ 称为 λ 的效果。从表 14.15 中可以看出，$k=2$ 时，有 3 个 λ：A，B 和 AB；$k=3$ 时有 7 个 λ：A，B，C，AB，AC，BC 及 ABC；$k=4$ 时有 15 个 λ：A，B，C，D，AB，AC，BC，AD，BD，CD，ABC，ABD，ACD，BCD 及 $ABCD$。

由于单因素及其交互效应的自由度为 1，所以其均方值为：

$$MS_\lambda = SS_\lambda$$

误差的均方值为：

$$MS_{error} = \frac{SS_{error}}{2^k(n-1)} \qquad n > 1$$

每一因素及其交互作用的检验统计定义为：

$$F_\lambda = \frac{MS_\lambda}{MS_{error}} = \frac{MS_\lambda}{SS_{error}/[2^k(n-1)]} \qquad \lambda = A, B, AB, \cdots \quad n > 1$$

2^k 析因实验的方差分析表如表 14.16 所示。

表 14.16　重复 $n>1$ 次的 2^k 析因实验的方差分析表

因素	平方和	自由度	均方值	F_λ	$f_{\alpha,1,(n-1)2^k}$
A	SS_A	1	MS_A	MS_A/MS_{error}	(f 表中的值)
B	SS_B	1	MS_B	MS_B/MS_{error}	(f 表中的值)
C	SS_C	1	MS_C	MS_C/MS_{error}	(f 表中的值)
⋮					
AB	SS_{AB}	1	MS_{AB}	MS_{AB}/MS_{error}	(f 表中的值)
AC	SS_{AC}	1	MS_{AC}	MS_{AC}/MS_{error}	(f 表中的值)
BC	SS_{BC}	1	MS_{BC}	MS_{BC}/MS_{error}	(f 表中的值)
⋮					
ABC	SS_{ABC}	1	MS_{ABC}	MS_{ABC}/MS_{error}	(f 表中的值)
⋮					
Error	SS_{error}	$2^k(n-1)$	MS_{error}		
Total	SS_{total}	$n2^k-1$			

2^k 析因设计的方差分析表可直接用于求多元回归模型,该模型对其过程的输出量进行估计。该过程的输出为主因素及其交互效应统计意义上显著性的函数。首先引入编码变量 x_β：

$$x_\beta = \frac{2\beta - \beta_{low} - \beta_{high}}{\beta_{high} - \beta_{low}}$$

其中,β 为主变量,即 $\beta = A, B, C, \cdots$。若 $\beta = A$,则当 $\beta = A_{high}$ 时,$x_A = +1$;而当 $\beta = A_{low}$ 时,$x_A = -1$。若 $A_{low} \leqslant A \leqslant A_{high}$,则 $-1 \leqslant x_A \leqslant +1$。

平均输出量 y_{avg} 的估计式为：

$$y_{avg} = \bar{y} + 0.5\Big[\sum_\lambda \text{Effect}_\lambda x_\lambda + \sum_\lambda \sum_\beta \text{Effect}_{\lambda\beta} x_\lambda x_\beta + \sum_\lambda \sum_\beta \sum_\gamma \text{Effect}_{\lambda\beta\gamma} x_\lambda x_\beta x_\gamma \cdots\Big] \tag{14.41}$$

式中,$\lambda, \beta, \gamma, \cdots$ 为 $A, B, C, \cdots$ 的值,且仅与那些表示显著性因素和显著交互效应的下标组合一致,$x_\alpha(-1 \leqslant x_\alpha \leqslant +1)$ 为编码值。

下面举例说明这些关系。

例 14.13　2^4 析因实验的分析

对表 14.17 中的数据建立方差分析表,表中数据为重复两次的 2^4 析因实验数据。在方差分析表中将包含效果。实验执行序号与表 14.14 中的一致。

首先创建函数 **FactorialData**,定义表中数据：

```
function dat = FactorialData
dat1 = [159 168 158 166 175 179 173 179 164 187 163 185 168 197 170 194]';
dat2 = [163 175 163 168 178 183 168 182 159 189 159 191 174 199 174 198]';
dat = [dat1 dat2];
```

表 14.17 重复两次的 2^4 析因实验数据——FactorialData

运行序号*	数据($y_{m,j}$)	
m	$j=1$	$j=2$
1	159	163
2	168	175
3	158	163
4	166	168
5	175	178
6	179	183
7	173	168
8	179	182
9	164	159
10	187	189
11	163	159
12	185	191
13	168	174
14	197	199
15	170	174
16	194	198

* 运行序号与表 14.14 的水平组合顺序一致。

然后,创建一函数,名为 **FactorialSigns**,该函数确定出表 14.15 中 k 个因素的符号,如下所示:

```
function s = FactorialSigns(k)
s = ones(2^k,2^k - 1);
for r = 1:2:2^k
 s(r,1) = - 1;
end
for c = 2:k
 e = 2^(c - 1);
 for r = 1:e
  s(r,e) = - 1;
 end
 for r = e + 1:2^(k)
  s(r,2^(c - 2)) = s(r - e,2^(c - 2));
 end
end
for m = 2:k
  e = 2^(m - 1);
  for j = 1:e - 1
      s(:,e + j) = s(:,j) .* s(:,e);
 end
end
```

生成方差分析表的程序如下:

```
tag = char('A','B','AB','C','AC','BC','ABC','D','AD','BD','ABD',...
        'CD','ACD','BCD','ABCD');
k = 4;n = 2;
fdata = FactorialData;
s = FactorialSigns(k);
Sm = sum(fdata')';
yBar = sum(Sm)/n2^k;
SStotal = sum(sum(fdata.^2)) - yBar^2 * n * 2^k;
for nn = 1:2^k - 1
 Clambda = sum(s(:,nn).* Sm);
 SSlambda(nn) = Clambda^2/2^(k + 1);
 EffectLambda(nn) = Clambda./2^k;
end
SSerror = SStotal - sum(SSlambda);
MSerror = SSerror/2^k;
f0 = SSlambda/MSerror;
pValue = 1-fcdf(f0,1,2^k);
disp('Factor      SS      MS      Effect      f-lambda      p-value')
disp([tag repmat('  ',15,1)num2str(SSlambda',6)repmat('  ',15,1)...
     num2str(SSlambda',6)repmat('  ',15,1)num2str(EffectLambda',6)...
     repmat('  ',15,1)num2str(f0',6)repmat('  ',15,1)num2str(pValue',6)])
disp(['SSerror 'num2str(SSerror,6)' 'num2str(MSerror,6)])
disp(['SStotal 'num2str(SStotal,6)])
disp(['yBar = 'num2str(yBar,6)])
```

其执行结果在 MATLAB 命令窗口中显示下列数据。

```
Factor   SS        MS        Effect    lambda      p-value
A        2312      2312      17        241.778     4.45067e-011
B        21.125    21.125    -1.625    2.20915     0.156633
AB       0.125     0.125     -0.125    0.0130719   0.910397
C        946.125   946.125   10.875    98.9412     2.95785e-008
AC       3.125     3.125     -0.625    0.326797    0.575495
BC       0.5       0.5       -0.25     0.0522876   0.822026
ABC      4.5       4.5       0.75      0.470588    0.502537
D        561.125   561.125   8.375     58.6797     9.69219e-007
AD       666.125   666.125   9.125     69.6601     3.18663e-007
BD       12.5      12.5      1.25      1.30719     0.269723
ABD      2         2         -0.5      0.20915     0.653583
CD       12.5      12.5      -1.25     1.30719     0.269723
ACD      0         0         0         0           1
BCD      0.125     0.125     0.125     0.0130719   0.910397
ABCD     21.125    21.125    -1.625    2.20915     0.156633
SSerror  153       9.5625
SStotal  4716
yBar =   175.25
```

由此看出,因素 A,C,D 及交互效应 AD 在大大高于 95% 的置信度上显著影响了过程的输出量。事实上,对这四个因素平方和求和,结果为 4485。因此,这些量平方和的不显著贡献为78 = 4716 - 4485 - 153,占总平方和的 1.65% 。

下面根据式(14.41)和上述结果求得置信度高于 95% 的回归方程:

$$y_{avg} = 175.25 + 8.50x_A + 5.44x_C + 4.10x_D + 4.56x_Ax_D$$

残差为测量值 $y_{m,j}$ 与 y_{avg} 间的差。其中，$y_{m,j}$ 与 y_{avg} 为表 14.14 中 x_γ 编码值的 2^4 种组合所对应的数据。下面编写程序计算残差，并用 normplot 函数在图中标出残差点。

```
fdata = FactorialData;
s = FactorialSigns(4);
yAvg = 175.25 + 8.5 * s(:,1) + 5.44 * s(:,3) + 4.1 * s(:,4) + 4.56 * s(:,1). * s(:,4);
normplot([fdata(:,1)-yAvg;fdata(:,2)-yAvg])
```

执行结果参见图 14.15。由图中看出，得到的残差值是可以接受的。

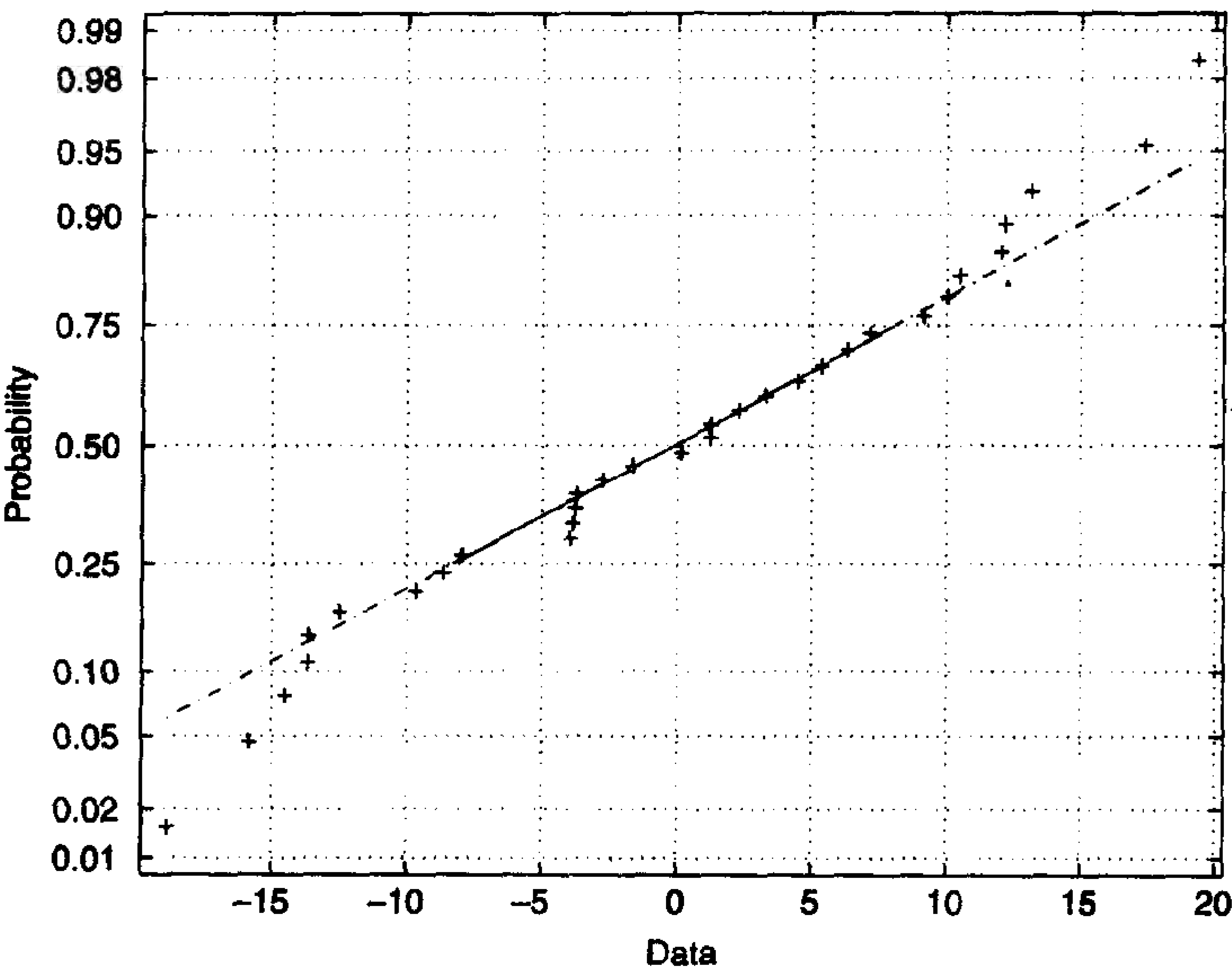


图 14.15 例 14.13 中所用数据的残差图

14.7 应用函数一览

表 14.18 列出了本章介绍的函数及其说明。

表 14.18 第 14 章介绍的 MATLAB 函数

MATLAB 函数	说明
anova1	单因素方差分析
anova2	双因素方差分析
binocdf	二项累积分布函数
binopdf	二项概率密度函数
binostat	二项分布的平均值和方差
boxplot	统计数据的盒状图
fcdf	*f* 累积分布函数
finv	*f* 累积分布函数的反函数
geomean	几何均值

（续表）

MATLAB 函数	说明
hist	画出直方图
histfit	直方图与正态分布函数的绘制
mean	平均值
median	中位数
normcdf	正态累积分布函数
normfit	正态数据参数和置信区间的估计
norminv	正态累积分布函数的反函数
normpdf	正态概率密度函数
normplot	正态概率图
poisscdf	泊松累积分布函数
poisspdf	泊松概率密度函数
poisstat	泊松分布的平均值和方差
polyconf	多项式计算和置信区间估计
prctile	采样的百分位数
prod	构建元素数组
range	数据范围
regress	多变量线性规划
skewness	采样偏移
std	标准方差
tcdf	t 累积分布函数
tinv	t 累积分布函数的反函数
ttest	单一样本均值的假设检验
ttest2	两样本均值差异的假设检验
wblcdf	韦布尔累积分布函数
wblfit	利用符合韦布尔分布的数据进行参数和置信区间的估计
wblinv	韦布尔累积分布函数的反函数
wblpdf	韦布尔概率密度函数
wblplot	韦布尔概率图
wblstat	韦布尔分布的平均值和方差

练习

14.1 (a) 一公司的热线帮助系统在工作时间内平均每分钟收到 5 个电话。求下列情况下，该公司能收到电话的概率为多少？(i)每分钟 8 个电话，(ii)每分钟 2 个电话。

(b) 电话系统每分钟能处理 10 个电话，如果多于 10 个，则打电话者将收到占线信号，求收到占线信号的概率为多少？

[答案：(a)(i)0.065 278，(ii)0.084 224，(b)0.013 695。]

14.2 一构件能承受负载 L_o 的概率为 0.7。如果用 15 个此类构件，问至少有 12 个能承

受负载 L_0 的概率为多少?

[答案:0.296 87。]

14.3 根据 Taguchi 的定义,过程的平均损失系数与下式成比例:

$$L_{avg} = s^2 + (\bar{x} - \tau)^2$$

式中为目标的平均值。换句话说,如果有两个过程,那么其中平均值最接近于 τ,方差最小的过程具有最低损失系数。对于表 14.19 所列数据,$\tau = 92.0$ 时,确定哪个过程具有最低平均损失系数。

[答案:$L_1 = 5.5904$, $L_2 = 2.6936$。]

表 14.19 习题 14.3 中的数据

过程 #1		过程 #2	
88.4	89.0	92.6	93.2
93.2	90.5	93.2	91.7
87.4	90.8	89.2	91.5
94.3	93.1	94.8	92.0
93.0	92.8	93.3	90.7
94.3	91.9	94.0	93.8

14.4 生产商发现有一种产品,其中的 20% 重量不足,这种产品一箱装有 24 件。如假定每件产品的重量与另一件产品的重量无关,则可以应用二项分布解决下列问题:

(a) 求装入箱子内重量不足产品的期望件数及其方差为多少?

(b) 在一箱内重量不足的产品不多于两件的概率为多少?

(c) 箱内没有重量不足产品的概率为多少?

(d) 在同一图上画出概率质量函数和累积分布函数随箱内重量不足产品数变化的曲线。

[答案:(a)期望值 = 4.8,方差 = 3.84;(b)0.114 52;(c)0.004 722 4。]

14.5 X^2 统计用于检验拟合优度。假定有 k 类(k 个单元),每一类的期望值用 $\hat{p}_i$ 表示,$i = 1,2,\cdots,k$。在一实验过程中,分布于每类的观察值(发生值)个数为 x_i,$i = 1,2,\cdots,k$,则 X^2 检验统计为:

$$\chi^2 = \sum_{i=1}^{k} (x_i - e_i)^2 / e_i$$

其中,$e_i = n = \hat{p}_i$,且:

$$n = \sum_{i=1}^{k} x_i = \sum_{i=1}^{k} e_i$$

如果任一类中的 $e_i < 5$,则必须将对应的 e_i 和 x_i 分别与其相邻的 e_i 和 x_i 组合直到 $e_i \geq 5$(参考习题 4.5)。

现用检验统计对以下假设做出检验:

$$H_0: p_i = \hat{p}_i \quad i = 1,\cdots,k$$

式中,$p_i = x_i / n$,如果:

$$\chi^2 \leqslant \chi^2_{\alpha,k-1}$$

则接受假设 H_0。实际中 α 未给定,而 p 值是给定的。p 值越接近于 1,越易相信观察得出的类发生值 x_i 与期望发生值 e_i 接近。

(a) 假定一种设备的故障发生概率如下所示:发生机械故障的概率为 0.60($\hat{p}_1=0.6$),发生电故障的概率为 0.25($\hat{p}_2=0.25$),由操作引起的故障概率为 0.15($\hat{p}_3=0.15$)。在记录的 55($n=55$)起设备故障中,有 32 起为机械故障($x_1=32$),14 起为电子故障($x_2=14$),9 起为操作故障($x_3=9$)。用上述方法判断值为 $\hat{p}_i$ 这台设备故障的期望值这一说法的合理性。

[答案:p 值 = 0.94979,因此是合理的。]

(b) 概率 $\hat{p}_i$ 也可以由统计模型得出。例如,假定有一基于 λ(假想值的)泊松分布,且:

$$e_i = n\frac{e^{-\lambda}\lambda^{i-1}}{(i-1)!} \quad i=1,2,\cdots,k+1$$

对 $i>k+1$,为:

$$e_i = n\left(1-\sum_{i=1}^{k+1}\frac{e^{-\lambda}\lambda^{i-1}}{(i-1)!}\right) \quad i>k+1$$

如果在容量为 85 的样本中,每件产品的故障次数如表 14.20 所示,则判断产品的故障次数服从 $\lambda=3$ 的泊松分布这一结论是否不合理。这里 $k=8$,而且需要对某些 e_i 和 x_i 进行重组。可以利用习题 4.5 的结果。

[答案:x_i = [17 20 25 14 6 3], e_i = [16.9276, 19.0436, 19.0436, 14.2827, 8.5696, 7.1330], p 值 = 0.405 92,所以结论有些合理。]

表 14.20　习题 14.5 中的数据

i	故障次数($i-1$)	发生次数 x_i
1	0	3
2	1	14
3	2	20
4	3	25
5	4	14
6	5	6
7	6	2
8	7	0
9	8	1
10	≥9	0

14.6 一部件的可靠率 $R(t)$ 为其在一段时间 t 内运行不失败的概率。如果部件寿命的概率分布函数为 $f(t)$,则其累积分布为:

$$F(t) = P(T \leqslant t) = \int_{-\infty}^{t} f(u)\mathrm{d}u = \int_{0}^{t} f(u)\mathrm{d}u$$

即运行直到失败时的概率。因此,

$$R(t) = 1 - F(t)$$

故障率函数 $h(t)$ 指部件在 t 时刻前未失败,在 t 时刻突然失败的偶然性,由下式

求出：

$$h(t)=\frac{f(t)}{R(t)}=\frac{f(t)}{1-F(t)}$$

(a) $f(t)$为一指数分布，且满足：

$$f(t)=\frac{1}{\mu}e^{-t/\mu}$$

要求 $\mu=1$ 用 exppdf 函数和 expcdf 函数在同一图上画出故障率函数和可靠率函数。

(b) 如果 $f(t)$服从威布尔分布，且 $\alpha=1,\beta=0.5,1,2$ 和 4，画出故障率函数和可靠率函数曲线。用 subplot 创建一含 4 个图形的(2×2)维数组。每一图形都包含对应一个 β 值的一对曲线。

14.7 对数分布的累积分布函数如下所述：

$$F(t)=\Phi\left(\frac{\ln(t)-\bar{x}_L}{s_L}\right) \tag{A}$$

其中，

$$\bar{x}_L=\frac{1}{n}\sum_{i=1}^{n}\ln(t_i) \qquad s_L^2=\frac{1}{n-1}\left[\sum_{i=1}^{n}(\ln(t_i))^2-n\bar{x}_L^2\right] \tag{B}$$

如果对累积函数求反函数，则得到：

$$y=\beta_0+\beta_1 x$$

式中，

$$y=\Phi^{-1}(F(t)) \qquad x=\ln(t) \quad \beta_0=-\frac{\bar{x}_L}{s_L} \quad \beta_1=\frac{1}{s_L}$$

$\Phi^{-1}(\cdots)$由 norminv 函数实现，t 的平均值和方差[不是 $\ln(t)$，$\ln(t)$服从正态分布]分别由下式求出：

$$\bar{x}_t=\exp(\bar{x}_L+s_L^2/2) \quad s_t^2=(\exp(s_L^2)-1)\exp(2\bar{x}_L+s_L^2)$$

上式可由 lognstat 函数求得。

(a) 对于表 14.21 中所列数据，利用例 14.3 所介绍的方法判断其是否服从对数分布。表中数据已排好序。另外，画出以 $\ln(t)$为自变量 $F(t)$函数的关系曲线以及拟合直线，并用 normplot 函数显示出方差值。

(b) 对用图形方法得到的 x_L 和 s_L 与由式(B)得到的 x_L 和 s_L 值比较。

[答案：从拟合曲线得：$x_L=2.5072$，$s_L=0.888\ 41$；由式(A)得：$x_L=2.5072$，$s_L=0.854\ 41$。]

表 14.21 习题 14.7 中的数据

1.55	15.70
3.05	16.35
3.65	17.70
5.20	17.95
7.75	19.45
10.45	19.80
10.85	20.05
10.90	32.75
12.65	35.45
15.25	49.35

14.8 过程能力比例(PCR, process capability ratio)是对某一过程是否满足由下限 LSL 和上限 USL 描述的一种度量。对于集中过程 PCR 定义为：

$$PCR = \frac{USL - LSL}{6\hat{\sigma}}$$

对于一非集中过程，PCR 定义为：

$$PCR_k = \min\left[\frac{USL - \bar{x}}{3\hat{\sigma}}, \frac{\bar{x} - LSL}{3\hat{\sigma}}\right]$$

为过程标准偏差的估计值，$\hat{\sigma}$ 为它和 $\bar{x}$ 平均值的估计值。当 $PCR > 1$ 时，几乎没有生产出坏部件或不合格部件；当 $PCR = 1$ 时，有 0.27%（每百万有 2700 个）不合格部件；当 $PCR < 1$ 时，生产出大量不合格部件。$100/PCR$ 为过程用到的定义宽度的百分值。当 $PCR = PCR_k$ 时，过程为集中的。

不合格部件数为 Np，其中 N 为生产部件的总量，且满足下式：

$$p = 1 - \Phi\left(\frac{USL - \bar{x}}{\hat{\sigma}}\right) + \Phi\left(\frac{LSL - \bar{x}}{\hat{\sigma}}\right)$$

式中，Φ 由式(14.16)定义，同时可参考图 14.5(c)和式(14.18)。

对表 14.22 中所列数据，用 MATLAB 的 capable 函数求出 p，PCR 和 PCR_k，已知 $LSL = 2.560$，$USL = 2.565$。问该过程是否为集中的？

[答案：$p = 1.5351e-004$，$PCR = 1.3103$，$PCR_k = 1.2099$。]

表 14.22　习题 14.8 中的数据

2.5629	2.5630
2.5630	2.5628
2.5628	2.5623
2.5634	2.5631
2.5619	2.5635
2.5613	2.5623

14.9 有一组数据如表 14.23 所示。下面假定两种情况：(1)表中所有数据为一集合，用 S_0 表示；(2)将列分为 5 组，每组数据单独为集合，用 S_j 表示，$j = 1, \cdots, 5$。

(a) 求 S_0 的调和平均值，并将其与算术平均值及几何平均值进行比较。调和平均值定义为 $\bar{x}_h$：

$$\frac{1}{\bar{x}_h} = \sum_{i=1}^{n} \frac{1}{x_i}$$

(b) 求 6 组数据各自的平均值及标准偏差 S_j，$j = 0, \cdots, 5$。

(c) 画出数据集 $S_j (j = 1, \cdots, 5)$ 的直方图。

(d) 求出 S_0 平均值与每个 $S_j (j = 1, \cdots, 5)$ 平均值之差的置信区间。置信度为 95%，假定标准偏差未知，但均相等(参考表 14.3 和表 14.5 中的情况 4)。每组数据的 p 值为多少？数据组 S_j，其平均值与 S_0 平均值是否存在显著性差异？定性分析所求结果与(c)中的图形是否一致？

表 14.23 习题 14.9 中的数据

1		2		3		4		5	
1115	1567	1223	1782	1055	798	1016	2100	910	1501
1310	1883	375	1522	1764	1020	1102	1594	1730	1238
1540	1203	2265	1792	1330	865	1605	2023	1102	990
1502	1270	1910	1000	1608	2130	706	1315	1578	1468
1258	1015	1018	1820	1535	1421	2215	1269	758	1512
1315	845	1452	1940	1781	1109	785	1260	1416	1750
1085	1674	1890	1120	1750	1481	885	1888	1560	1642

14.10 为确定能否使用表 14.5 中的情况 4 或情况 5,首先要对情况 7 中表示的方差比值进行 F 检验。如果方差之间显著不同,则可用情况 5 分析;否则用情况 4 分析。要求编写一段程序确定表 14.24 中两组数据平均值之间是否存在差异,根据程序运行结果确定这一点,并形成一盒状图,从图中定性分析上述结论是否正确。
[答案:对方差比值做 F 检验,结果 $p = 0.470\,92$,因此方差之间没有差异。对平均值之差做 t 检验,结果 $p = 0.000\,934\,2$,因此平均值间有差异。]

表 14.24 习题 14.10 的数据

组 1		组 2	
88	81	76	79
79	83	83	85
84	90	78	76
89	87	80	80
81	78	84	82
83	80	86	78
82	87	77	78
79	85	75	77
82	80	81	81
85	88	78	80

14.11 (a) 用下列模型对表 14.25 所列数据求 β_j,并说明该模型能很好地拟合表中数据。

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

(b) 用求得的 β_j 画出曲面图,并用 contour 函数画出该曲面在(x_1, x_2)平面上的等高线投影。

(c) 求出曲面的极大值坐标。

[答案:$x_1 = 18.7635, x_2 = 38.0156$。]

表 14.25 习题 14.11 中的数据

y	x_1	x_2
144	18	52
142	24	40
124	12	40
64	30	48
96	30	32

（续表）

y	x_1	x_2
74	26	56
136	26	24
54	22	64
92	22	16
96	14	64
92	10	56
82	10	24
76	6	48
68	6	32

14.12　一元线性回归分析的相关系数 R 可由式(14.29)求出，现对以下假设做出检验：

$$H_0: R = 0$$

$$H_1: R \neq 0$$

建立检验统计量：

$$t_0 = \frac{R\sqrt{n-2}}{\sqrt{1-R^2}}$$

将 t_0 与 $t_{\alpha/2,n-2}$ 进行比较，如果 $t_0 > t_{\alpha/2,n-2}$，则放弃 H_0。实际检验的是对应 t_0 的 p 值。$n \geqslant 25$ 时，相关系数 r 的置信区间由下式求出：

$$\tanh\left(\tanh^{-1}(R) - \frac{-z_{\alpha/2}}{\sqrt{n-3}}\right) \leqslant r \leqslant \tanh\left(\tanh^{-1}(R) + \frac{z_{\alpha/2}}{\sqrt{n-3}}\right)$$

式中，$z_{\alpha/2}$ = norminv$(1-\alpha/2)$[参考式(14.19)]。

对于表 14.26 所列数据：

(a) 如果回归模型为 $y = \beta_0 + \beta_1/x$，确定回归系数。

(b) 画出回归线及各个数据点。

(c) 判断残差是否为正态分布。

(d) 确定(i)相关系数是否等于 0，(ii)置信度为 95% 的置信区间。

[答案：(a) $\beta_0 = 8.9366, \beta_1 = -41.6073$；(d) $0.976\,998 \leqslant 0.989\,96 \leqslant 0.995\,64$。]

表 14.26　习题 14.12 中的数据

x	y	x	y
10.0	4.746	11.6	5.211
12.0	5.466	14.8	6.264
6.8	3.171	7.2	3.411
5.4	1.500	15.7	6.537
20.0	6.708	17.6	6.336
19.4	7.158	14.0	5.400
19.1	6.882	10.9	4.503
6.1	1.674	18.2	6.909

(续表)

x	y	x	y
16.3	6.498	20.4	6.930
12.4	5.598	8.2	3.582
5.8	1.959	7.9	3.432
12.7	5.790	4.9	0.369
9.2	4.686		

14.13 多元线性回归分析中，经常检验的两类残差为：

(a) 标准化残差，其定义为：

$$d_i = \frac{e_i}{\hat{\sigma}}$$

其中，$\hat{\sigma}^2$ 由式(14.38)定义。

(b) 学生化残差，定义如下：

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$$

式中，h_{ii} 为下式的第 i 个对角线元素：

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$$

$\boldsymbol{X}$ 由式(14.32)定义。

用例 14.10 中的模型及表 14.7 中的数据求出其标准化残差和学生化残差。$\hat{y}_i$(即 $y_i - e_i$，y_i 在同一图上画出这两种残差随平均输出量为表 14.7 中的输出值)的变化图。用不同符号区别两幅图形。分析这些方差值有无远距离点。在此，超过 3 就属于远距离点。用 legend 函数画出图并标出两组不同的残差值。

14.14 2^k 析因实验公式在 $n=1$ 时可做如下修改：效果 Effects_λ 的计算如 14.6.2 节所示，将其按从正的最大值到负的最大值进行降序排序，参考例 14.13 结果表中标为"Effect"的列。排序过程中必须记录所有因素及其交互效应。已排序的效果用 normplot 函数在图上画出。可以忽略的效果(不显著的)将服从正态分布，在图中分布于一条直线上，而那些显著性效果值将明显偏离该直线。表 14.27 中的数据为重复一次的 2^4 析因实验所得数据，确定其中哪些值为显著的。可以从 normplot 函数得到的图中直观地看出，也可由已排序的效果及其对应的因素或交互效应的列表中直观地看出。

[答案：A, C, D, AC, AD 占总平方和的 96.6%。对因素和交互效应进行由正到负的排序：A, C, B, BCD, BC, ABC, ACD, CD, BD, AB, $ABCD$, ABD, D, AD 和 AC。$\text{Effect}_A = 43.25$，$\text{Effect}_C = 19.75$，$\text{Effect}_D = -29.25$，$\text{Effect}_{AD} = -33.25$，$\text{Effect}_{AC} = -36.25$。]

表 14.27 重复 $n=1$ 次的 2^4 析因实验数据

运行序号⁺	y_m	运行序号⁺	y_m
1	86	9	90
2	200	10	142
3	90	11	96

(续表)

运行序号⁺	y_m	运行序号⁺	y_m
4	208	12	130
5	150	13	136
6	172	14	120
7	140	15	160
8	192	16	130

⁺运行序号与表 14.14 中的水平组合相一致。

参考文献

1. T. B. Barker, *Quality by Experimental Design*, Marcel Dekker, New York, NY, 1985.
2. G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters*, John Wiley & Sons, New York, NY, 1978.
3. F. W. Breyfogle III, *Statistical Methods for Testing, Development and Manufacturing*, John Wiley & Sons, New York, NY, 1992.
4. N. Draper and H. Smith, *Applied Regression Analysis*, 2nd ed., John Wiley & Sons, New York, NY, 1981.
5. E. A. Elsayed, *Reliability Engineering*, Addison Wesley Longman, Inc., Reading, MA, 1996.
6. N. L. Frigon and D. Mathews, *Practical Guide to Experimental Design*, John Wiley & Sons, New York, NY, 1997.
7. A. J. Hayter, *Probability and Statistics for Engineers and Scientists*, PWS Publishing Co., Boston, MA, 1996.
8. L. L. Lapin, *Modern Engineering Statistics*, Duxbury Press, Belmont, CA, 1997.
9. E. E. Lewis, *Introduction to Reliability Engineering*, 2nd ed., John Wiley & Sons, New York, NY, 1996.
10. D. C. Montgomery, *Design and Analysis of Experiments*, 3rd ed., John Wiley & Sons, New York, NY, 1991.
11. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, New York, NY, 1994.
12. R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, NY, 1995.
13. R. E. Walpole, R. H. Myers, and S. L. Myers, *Probability and Statistics for Engineers and Scientists*, 6th ed., Prentice Hall, Upper Saddle River, NJ, 1998.